An Agent-Based Approach to Spatial Epidemics through GIS

Joana Margarida de Almeida Simões

Centre for Advanced Spatial Analysis & Department of Geography

University College London

University of London

A thesis submitted for the Degree of Doctorate of Philosophy at the University of London

September 2006

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

I, Joana Margarida Almeida Simoes, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Signed:.....

ABSTRACT

This study introduces a bottom up approach to model the complex phenomena involving the spread of epidemics in a human population with spatial dependence. To pursue this, several concepts and technologies are assembled together such as Artificial Life, Complex Networks, and Geographical Information Systems (GIS).

At the beginning of this project, a Cellular Automata (CA) model was developed, but that was replaced later by an agent-based model (ABM) as it seemed more adequate for this particular context; the main target was relaxing the assumptions of homogeneous space and of random mixing of individuals. The model is split into a movement model and an infection model. The movement model uses network concepts and combines different ranges of movements to emulate the movement of individuals in human populations. The infection model is based on a classic SEIR (Susceptible-Latent-Infectious-Removed) model and it is flexible enough to accommodate different infectious diseases. Both these models are probabilistic. The model was implemented from scratch in C++, and developed within a Geographical Information Systems (GIS) environment that provides functions for the display and analysis of geographic information.

Finally, some spatial vaccination strategies were implemented which illustrates a useful purpose of this kind of simulation model, for instance for aiding in the definition of health policies.

The documented simulations cover the sensitivity analysis of the parameters of the movement and infection model, and there are also simulations to reproduce a mumps outbreak that occurred in Portugal between 1993 and 1996. In this way, an application of the epidemic model was illustrated for a case study, that enabled an analysis and interpretation of the model efficiency.

ACKNOWLEDGMENTS

A work like this can only be achieved with the help and support of a large group of people and institutions. This section may be extensive and tedious but since the people who actually bother to read it, will look for their names in it I am going to make an effort to not leave anyone aside.

First of all, I would like to thank my Supervisor, Michael Batty, for his great support and encouragement at all moments, and also for his scientific advice regarding the complex world of *hi-tech* Geography. It was a great joy to have such close contact with a person whose contributions were, and still are, so important in this field and his multidisciplinary views were most appreciated. I also have to thank him as the director of CASA (Centre for Advanced Spatial Analysis), for hosting me all these years and giving me the opportunity of developing my research in such an environment! I use the opportunity to thank as well the CASA staff, including the deputy director Paul Longley who is also my second supervisor, and my PhD colleagues specially those who started with me and shared the Chandler House office for almost three years (Aidan, John, Ana, Andrew,...)! I am happy some of them became my friends.

There were two books which really motivated me for this work. One was *Mapping Cyberspace* from Martin Dodge, who I had the chance to meet personally, that put me in contact with CASA and the kind of research developed there. The other was *Foundation* from Isaac Asimov, who I did not have the chance to meet personally, which showed me that exact and social sciences can be successfully combined together in a bottom up approach. I would like to thank these two authors for their inspiration!

For motivating me into the field of Artificial Life and allowing me such great and productive discussions, I have to thank the Artificial Life Reading Group (Alergic) from the University of Sussex, specially Ezequiel A. Di Paolo for his great encouragement and kind invitations for talks, which were most appreciated.

There were also people who gave specific contributions, critically reading parts of my

thesis that were inside their expertise. One was Genaro Martnez, who kindly looked at my Cellular Automata stuff and mathematical notation and also discussed with me some ideas about this field. The other was Carlos Pinto who read the implementation technical stuff and also discussed with me generic Artificial Life concepts, although we never agree on anything! (that argument on the plane was great!) My big *Thanks* to both of them and also to Paulo Morgado, who patiently produced the nice framing maps I have in the *Results* Chapter.

From Portugal, I have to mention one person as really important, although we did not keep so much in touch throughout the thesis. This person is Carmo Gomes from FCUL, who patiently talked to me when I was nearly starting the thesis and hardly knew anything about epidemics. He also kindly read my chapter about the biology of the disease and most important of all, provided me with the dataset I use for my simulations which is of a great value in a time when is so difficult to get data for research purposes! I am deeply grateful for all of this.

Since I was very lucky in my work environments I have to thank my previous employer, Manuel Afonso Dias, for all his support before and during the thesis and also my current employer for his support, his help and for giving me the opportunity of a challenging work that will pursue, develop and make useful the research I started in this PhD. Both of them are people with a great vision of scientific research, which I share, emphasizing its value for society.

I want to thank José António Tenedório for encouraging with my research but mostly for giving me the rare opportunity to make my work known in Portugal and keep a research link with my country, through the organism e-GEO. This was most appreciated!

I express here my gratitude towards two organisations whose financial help was essential for this project: the Portuguese Foundation for Science and Technology (FCT) and the Economic and Social Research Council (ESRC). The grants they awarded me, for living expenses and fees, made my life in London and UCL possible. Thank you for believing in me and giving me this opportunity.

Now coming into a different note, I would like to thank:

- The Fairtrade coffee producers, since programming is a way of turning coffee into code and I did a lot of programming in this PhD! I support unconditionally the *Fairtrade* movement because I think it is a small gesture than can make a difference in this globalized world and also because they offered me a beautiful mug.
- All the music I have been listening to, while I am working. Funny enough, the top musicians when I am programming are Mafalda Veiga and Jack Johnson and different kinds of Jazz and Bossa Nova. A big *Thanks* to all of them for keep my mind productive!
- Linux, and therefore Linus Torvards, for making my life so complicated but fascinating at the same time! I also have to express my gratitude towards the *gurus* that gave me some kind of technical support: my brother João and my friend Cristina. When you are in deep Linux trouble, these are the people you want to ask for help!

As I am very lucky to have such good friends, I could not forget to mention their support and the happy moments we spent together that helped me keeping my sanity, even in more stressful times. Since I lived in different cities, I have different groups of friends but I am happy that even with distance or other problems, we managed to keep - and even make stronger! - the ties between us; from Lisbon: Luis, Cristina, Dora (and everyone from *Moinho*!), Ana, Zé-Carlos and Teresa Gabriel, just to mention some; from Faro: Carlos, Elsa, Hugo, and everyone I met and spent time with when I was living there. from London: Jana (*danke schon!*), Sofia, Inge, Monica, Gail and other people that are not here anymore or that I met recently (Enrique and all the spanish people!). Thank you so much for keeping me busy (and entertained!) with your visits, even when I was really busy working on my PhD!

Finally, I have to say *thank-you* to my family, particularly my Mum and Dad, without whom this PhD would not have been possible. Their support, encouragement and love, at all moments, has a value that I am not able to express in words. I also thank my

grandmother for caring so much about me and feeding me with the best honey cakes in the world! My brother, apart from being my Linux *guru* was also the person who most encouraged me into programming and it is due to him that I turned from a Visual Basic for Applications programmer to a C++ developer! Is that all? no, I am forgetting my cats (the three of them!)

Once again, **thank you** for making this possible and I would like to dedicate this thesis to ALL OF YOU.

Table of Contents

List of Tables

Listings

List of Figures

Abbreviations

ABM	Agent-Based Modelling
AGI	Artificial General Intelligence
AI	Artificial Intelligence
AIDS	Acquired Immunodeficiency Syndrome
ALife	Artificial Life
API	Application Program Interface
CA	Cellular Automata
CNIG	Centro Nacional de Informacao Geografica
COM	Component Object Model
CPU	Computer-Processing Unit
DBMS	Database Management System
DTM	Digital Terrain Model
EPISIM	Epidemic Simulation
FHP	Frisch, Hasslacher and Pomea
FSP	Finite State Machine
GA	Genetic Algorithms
GIS	Geographic Information Systems
GNU	GNU's Not Unix
GPL	General Public License
HIV	Human Immunodeficiency Virus
HPP	Hardy, de Pazzis and Pomeau
INE	Instituto Nacional de Estatistica
I/O	Input/Output
LGCA	Lattice Gas Cellular Automata
ODBC	Open DataBase Connectivity
OCX	OLE Control Extension

OLE	Object Linking and Embedding
OO	Object Oriented
SARS	Severe Acute Respiratory Syndrome
SECA	Site Exchange Cellular Automata
SW	Small Worlds
TIN	Triangular Irregular Network
UI	User Interface
VB	Visual Basic
VBA	Visual Basic for Applications

References to Programs and Libraries

GNU C++ This C++ compiler is part of the GNU Compiler Collection, produced by the GNU Project. It is a free software and it has been used as the standard compiler for the free software Unix-like operating systems and the latest versions of Mac OS X. `http://gcc.gnu.org/`

GRASS GRASS stands for Geographic Resources Analysis Support System and it is an open-source, free, GIS software, released under the GNU license. It provides support for raster, topological vector, image processing, and graphics production and operates on various platforms although it is native in the X Window System. `http://grass.itc.it/`

MapObjects — This commercial toolkit from ESRI consists in a collection of embeddable mapping and GIS components, for Windows developers. `http://www.esri.com`

Oracle9 — This refers to the management system software used to create an Oracle relational database. The software is produced and marketed by Oracle Corporation and it is widely used in many database applications on many computing platforms. `http://www.oracle.com/index.html`

Quantum GIS — Quantum GIS, also called QGIS, is a user friendly Open Source Geographic Information System (GIS) released under the GNU license. QGIS supports vector, raster, and database formats and it has the advantage of running in different plattforms. `http://qgis.org/`

Qt — Qt is a a cross platform, graphical widget, C++ toolkit. Although it is mainly used for the development of GUI programs, the latest versions also provide a lot of non-GUI features. Trolltech, which produces this software, releases it under a dual license, the GPL and proprietary Q Public License, on all supported platforms. `http://www.trolltech.com/`

Shapelib — This is a simple C library that provides the ability to read, write and update ESRI Shapefiles, and the associated attribute file (.dbf). `http://shapelib.maptools.org/`

TOra This is an open-source toolkit which aims to help the developer of an Oracle database application. TOra also includes support for MySQL and Post. `http://sourceforge.net/projects/tora/`

VB Visual Basic is an event driven programming language associated to the development environment from Microsoft. It is derived from BASIC and enables the rapid application development of graphical user interface applications `http://msdn2.microsoft.com/en-us/vbasic/default.aspx`

"Art is a lie that enables us to realize the truth."

Picasso, cited in Light (2000)

Chapter 1

Introduction

1.1 Motivation and Background

The primary motivation for this study was to develop a bottom-up approach to modelling spatial diffusion phenomena. In this framework, the spatial component of the model was as important as the temporal one, and therefore the relevance of using Geographic Information System (GIS) technology is central to this thesis. As this study stands at the intersection of several fields as is shown in figure 1-1, different areas are introduced through the literature review; the GIS topic is not discussed in this chapter for we will develop it as an entire section in chapter 3 (section 3.4, page 92).

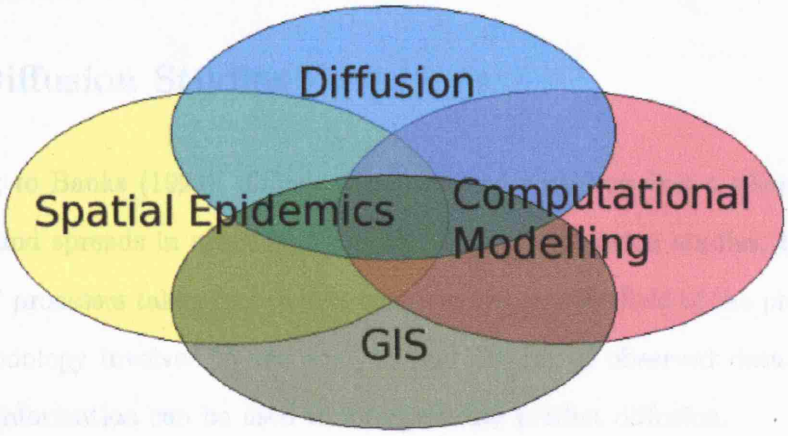


Figure 1-1: The multidisciplinary scope of this study.

Besides building a model and implementing it (thus bringing in some technical issues), this study also wanted to cover the adversities we have when dealing with real data, rather than a theoretical and perfectly controlled dataset. In this sense, the application to a case study can be seen as an exercise concerning the process of integrating real data in an abstract, non-data driven model, rather than a validation of the model. Moreover this model aims to be a tool for understanding the mechanisms of disease spreading, and also in more general terms for bringing methodological ideas to the study of spatial diffusion.

This thesis is structured in the following way: it starts with a literature review that introduces the theme and what has been accomplished so far in the subject in

chapter 1 (page 26). Then in chapter 2 (page 49), it describes the conceptual model from a theoretical point of view and chapter 3 (page 75) explains how the model was implemented, approaching technical questions such as programming and GIS. Chapter 4 (page 116) is concerned with a sensitivity analysis of the model parameters through several simulations but at this point, not focusing on real data. The application to a case study using a dataset from a mumps outbreak in Portugal (1993-96) is described in chapter 5 (page 160). The following chapter - chapter 6 (page 176) - is entirely dedicated to vaccination strategies. Finally, conclusions and further developments are presented in chapter 7 (page 195).

1.2 Diffusion Studies

According to Banks (1994), diffusion is concerned with how fast a phenomena grows, transfers and spreads in space and through time. In diffusion studies, the question is not "why" processes take place (which concerns the specific field of the phenomena) but the methodology involved in the analysis and display of observed data, and how the resulting information can be used to interpret and predict diffusion.

There is an extensive mathematical framework for studying diffusion, which is expressed in simple relations such as exponential functions or growth relationships as in equation in (1.1), or more complicated forms such as the standard diffusion equation in (1.2).

$$\frac{dN}{dt} = \sigma N \tag{1.1}$$

$$\frac{\partial N}{\partial t} = D\frac{\partial^2 N}{\partial x^2} + \sigma N \tag{1.2}$$

These equations explain the variation of the magnitude of a growing quantity (N) with time (t) through the intrinsic growth coefficient (σ). In the case of (1.2), a spatial dimension (x) is considered and this introduces the diffusion coefficient (D).

Spatial diffusion is a more specific type of diffusion that can be defined as the prop-

agation of phenomena in time and space, of simple or complex elements (Dauphiné, 1995). There are several classifications of spatial diffusion, for instance according to its scale or the nature of its processes (Dauphiné, 1995); Cliff et al. (1981) present a classification based on the behavior of diffusion processes which is represented in figure 1-2).

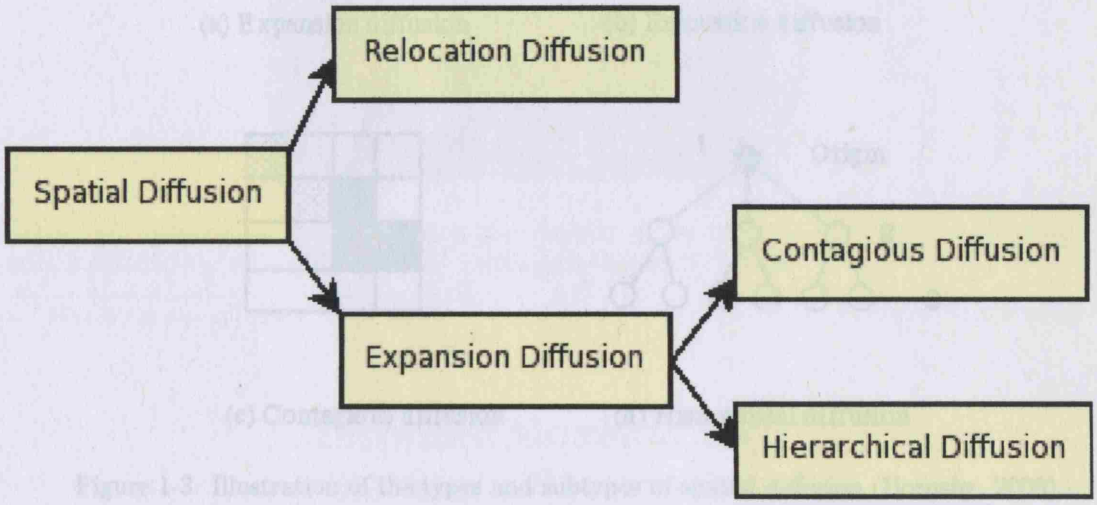
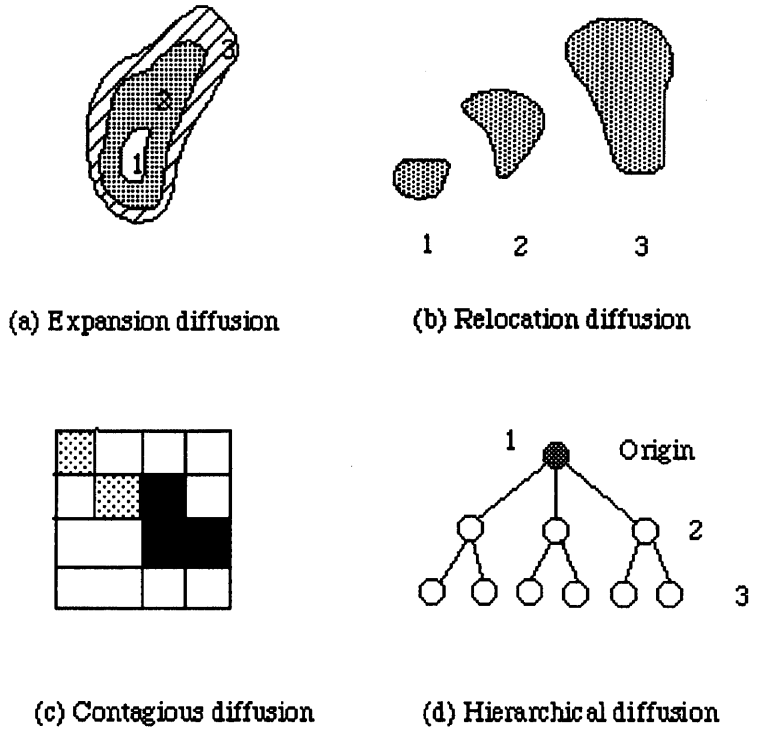


Figure 1-2: Spatial diffusion processes (based on Cliff et al. (1981)).

This classification distinguishes between relocation diffusion and expansion diffusion, the first representing a phenomena where an element moves along with the process, and the second representing a phenomena where traces of the process are left behind, as shown in figure 1-3. Two typical examples of these types of diffusion are migration (relocation diffusion) and cultural innovation (expansion diffusion). Expansion diffusion can be subdivided into contagious diffusion where the phenomena spreads evenly by spatial proximity, and hierarchical diffusion where the space is split in units of different importance and the phenomena spreads from more important units to less important ones. A good example of contagious diffusion is an epidemic, while hierarchical diffusion can be exemplified by the spread of cultural innovations, from one important person to another or from one urban center to another, temporarily bypassing persons of lesser importance and rural areas (Jordan et al., 1994).

In nature and human activity, there are several examples of spatial diffusion phenomena: the spread of a wildfire, adoption of herbicides, migration, oxygen transfer across



(a) Expansion diffusion (b) Relocation diffusion

(c) Contagious diffusion (d) Hierarchical diffusion

Figure 1-3: Illustration of the types and subtypes of spatial diffusion (Hornsby, 2006).

a water surface and so on; two illustrative examples are shown in figures 1-4 and 1-5.

There were several reasons that lead to the choice of human epidemics as the target of this study. First of all, this study aims to implement a bottom-up approach including behaviors such as emergence and self organization and in this sense a complex system such as the one where human epidemics can take place is a good example (see section 1.4, page 33). Secondly, epidemics repeat themselves in time and space and so there is an ability to forecast spread. Even if eradication is not possible, vastly reduced levels of disease can be achieved if we can forecast the time-space incidence of the disease and target medical assistance (Cliff et al., 1995). Also, recent virus epidemics such as the Severe Acute Respiratory Syndrome (SARS) have reached the attention of governments, public health officials, and the media due to the importance of predicting and controlling the patterns of viral infection. Besides the unquestionable value of human life, decreased worker productivity as a result of illness costs industry millions of dollars every year (Conference of the First Australian Conference on Artificial Life 2003, 2003), and thus a better understanding of such infections is fundamental. Lastly,

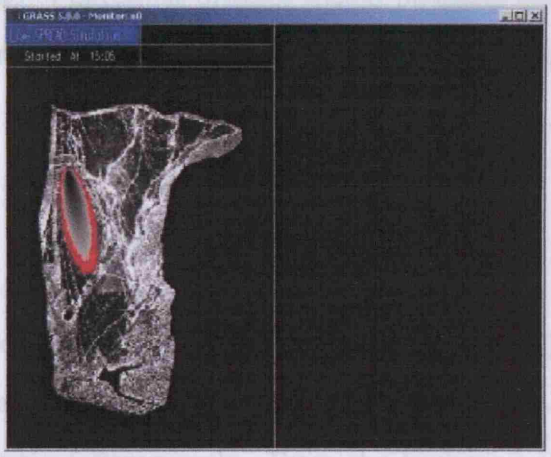


Figure 1-4: Example of spatial diffusion phenomena: spread of a wildfire; this is a screenshot from a simulation run on GRASS GIS[1] , with demo model SPREAD.

[1] http://grass.itc.it

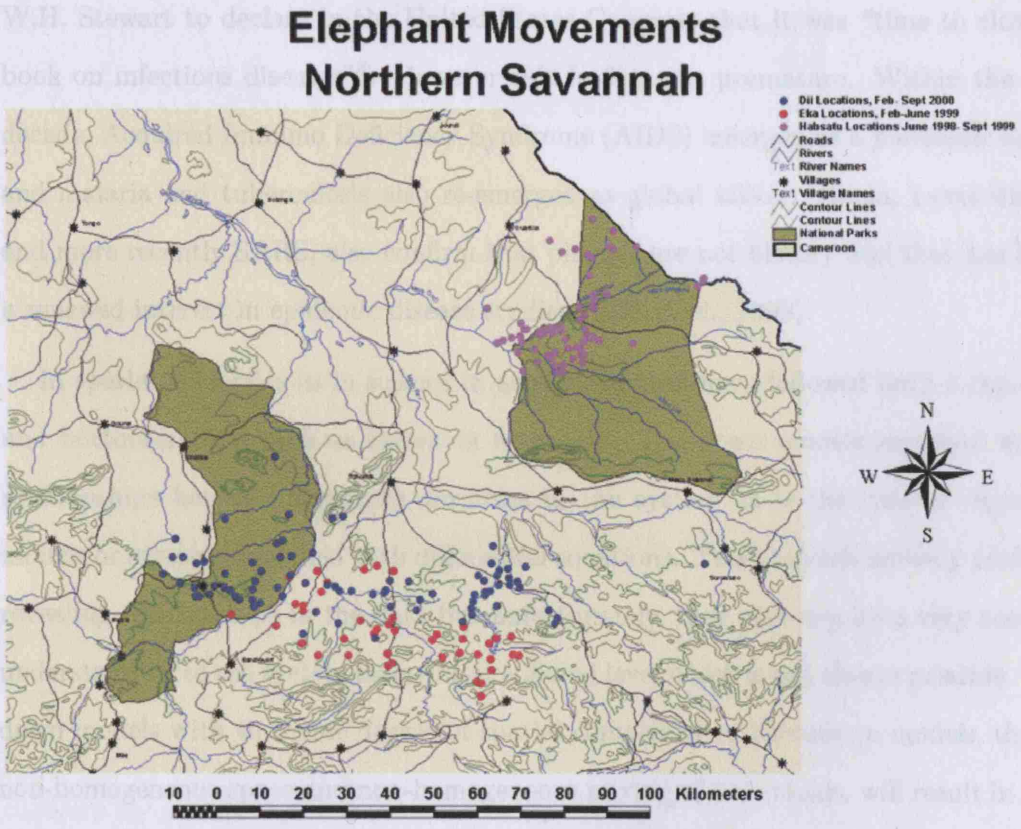


Figure 1-5: Another example of spatial diffusion phenomena: migration of elephants[2]. The map shows the movements of three animals near Bouba Njida and Benoue national parks (near N'Gaoundal and Cameroon).

[2] http://www.fieldtripearth.org/print-article.xml?id=775

models of the spread of infectious diseases should be useful generally in the analysis of other diffusive phenomena such as innovation or cultural diffusion patterns (Cliff et al., 1981).

1.3 Spatial Epidemics

Spatial epidemics can be defined according to Lawson (2001) as the analysis of the spatial/geographical distribution of the incidence of a disease - where we use the term spatial and geographic synonymously. From the beginning of the twentieth century till its last decades, there have been great falls in mortality from classical infectious diseases, due mostly to the successes of global campaigns against smallpox and malaria. In 1970, the sense of security brought by the advent of vaccines and antibiotics lead W.H. Stewart to declare in the United States Congress that it was "time to close the book on infectious diseases"[3]. However this feeling was premature. Within the same decade, Acquired Immuno Deficiency Syndrome (AIDS) emerged as a pandemic disease and malaria and tuberculosis also re-emerged as global killers. Ebola, Lyme disease, and more recently SARS, also confirm that plagues are not history and thus has led to a renewed interest in epidemic disease studies (Cliff et al., 1998).

In spatial epidemics as in science in general, models have followed both a top-down and bottom-up approach as shown in figure 1-6. Top-down models represent system relationships between aggregate variables in the system as is the case of regression models or systems modelled with differential equations. These models are very useful for revealing relationships in the data but unfortunately, they also require a very accurate understanding of the system behavior at a global level which is not always possible. Top-down models with the same degree of spatial complexity of bottom-up models, that is, non-homogeneous space and non-homogeneous mixing of individuals, will result in large systems of partial differential equations that are too complex to be solved analytically. The numerical solution of these equations is computationally intensive. Bottom-up models start from a general understanding of the low-level processes and their elements,

[3] http://www.fda.gov/fdac/features/396_infe.html

and generate aggregate system behavior by simulating the individual entities in the system. Most epidemic models found in the literature fall in the category of top-down models and these will not be described in the context of this work. Recently, following the evolution of computational modelling of spatial epidemics, bottom-up models have increased in number and quality. In the next section, we will explain why we believe this kind of approach is more suitable for the study of spatial epidemics and some key concepts will be introduced. Following that, there will be a short review of what has been done so far in the field and this will introduce this study in its context.

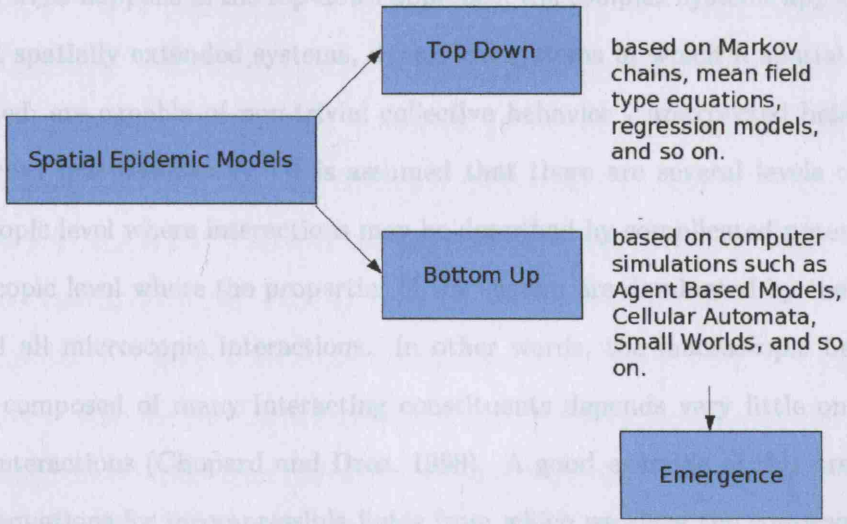


Figure 1-6: Approaches in spatial epidemic modelling.

1.4 Complexity Theory and Computational Modelling

Complexity Theory can be defined as the study of systems that are composed of of a large number of (often simple) components (Cilliers, 1998). These components interact dynamically in a non-linear way and therefore the system cannot be described accurately by a limited set of abstract rules. However, that does not mean that the structure of complex systems is arbitrary, or chaotic. Human-environment systems are characterized by heterogeneity, nonlinear relationships, and hierarchical structures that give rise to difficulties in understanding system behavior in response to exogenous fac-

tors (8th International Conference on GeoComputation, 2005). In this sense, they are the perfect target for a complex systems approach. It is important to note that human epidemics are strongly related to the dynamics of populations and therefore to the network of social contacts. These kinds of systems manifest **self organization**, a behavior in which order may arise from low level interactions without any supervision from higher-order structures (Nowak and Lewenstein, 1996). Self organizing systems typically display a behavior called **emergence**, which is the process of formation of complex and unexpected high-level patterns, from simple rules (Chalmers, 1996). Contrary to what happens in the top-down approach, the complex systems approach considers that spatially extended systems, dynamical systems in which n spatial dimensions are added, are capable of non-trivial collective behavior - unexpected behavior which is observed macroscopically. It is assumed that there are several levels of reality: a microscopic level where interactions may be described by complicated potentials, and a macroscopic level where the properties of the system are dominated by the aggregated effect of all microscopic interactions. In other words, the macroscopic behavior of a system composed of many interacting constituents depends very little on the details of the interactions (Chopard and Droz, 1998). A good example of this are the Navier Stokes equations for incompressible fluids from which we show the conservation of momentum in (1.3). The details of the microscopic interactions are present in its coefficient (viscosity) but have no expression in its algebraic structure.

$$\frac{\partial u}{\partial t} + u.\nabla u = -\frac{\nabla P}{\rho} + \nu \nabla^2 u \tag{1.3}$$

In this equation ν is the kinematic viscosity, P is the pressure and ρ is the fluid density. ∇u is the Laplacian of the velocity of the fluid parcel which is the vector containing the equations for the three components of velocity in a Euclidean Space (x,y and z).

Once emergent properties cannot be trivially derived from the properties of individual elements, it is difficult to predict them. Thus the use of computer simulations allows

the precise study of the dynamical consequences of models which cannot be solved by analytical methods, and therefore computation is of central importance in this kind of approach.

Moving to the field of computation, it is important to introduce two concepts which are commonly confused: **Artificial Life** (ALife) and **Artificial Intelligence** (AI). Both ALife and AI research are focused on understanding the properties of living organisms so that they can build artificial systems that exhibit these properties for analytical purposes. The difference is that AI is mostly focused on perception, cognition and generation of action, whereas ALife is focused on evolution, reproduction, morphogenesis and metabolism. Both share an interest in synthesizing adaptive autonomous agents; that is, computational systems that inhabit some complex, dynamic environment, sense and act autonomously in this environment and by doing so, realize a set of goals or tasks that they are designed for[4]. It could be argued that the last sentence, although valid for Artificial General Intelligence (AGI), does not apply to all AI. Because the mechanisms of intelligence at a general level are barely known, AI has focused in domain-specific "intelligence" such as pattern recognition, expert systems or optimization algorithms. By splitting the complexity of intelligence in small processes, AI managed to reduce its complexity but by adopting a top down approach it is using a methodology for generating intelligent behavior that has no relation to the way intelligence itself is generated in natural systems. We can say that AI is primarily focused on producing *intelligent solutions* rather than *intelligent behavior* (Langton, 1989). On the other hand, most mechanisms by which the ALife behaviors in real systems arise are well known. Therefore ALife is not focused on building systems that reach some sort of solution, but in the ongoing dynamics of the systems themselves. ALife starts at the bottom, viewing an organism as a large population of simple *machines*, and works upwards synthetically, constructing large aggregates of simple rule based objects which interact nonlinearly in the support of global dynamics (Langton, 1989). The purpose of this methodology is to obtain emergent behavior: natural life emerges out of the organized interactions,

[4] `http://www.aaai.org/AITopics/html/alife.html`

with no global controller responsible for the behavior of every part. There are two main approaches to ALife. The strong ALife position states that *life is a process which can be abstracted away from any particular medium* (Neumann, 1996); one example of this would be the program *Tierra*[5] which is not meant to simulate life in a computer, but to synthesize it. The weak ALife position denies the possibility of generating a *living process* outside of a carbon-based chemical solution. Instead, its researchers try to mimic life processes to understand the appearance of single phenomena. One example of this is the Agent-Based Model (ABM). The field of ALife is also characterized by the extensive use of computer programs and computer simulations which include evolutionary computation (Evolutionary Algorithms (EA), Genetic Algorithms (GA) and Swarm Intelligence (SI)), agent-based models, and Cellular Automata (CA)[6].

However these are not the only approaches. Complex systems are networks of all kinds, ranging from human movements, nervous systems, cell transfers, social behaviors, and so on, and it makes sense to look at this field in network terms. In network science, a **complex network** can be defined as a network or graph that has non-trivial topological structure[7]. Examples of topological features that do not exist in simple networks (or random graphs) are those heavy-tails which appear in the degree distribution, high clustering coefficients, assortativity or disassortativity among vertices, community structure at many scales and evidence of a hierarchical structure. Two well known models of complex networks are the scale free network and Small-World (SW) network that will be be described in more detail on section 2.2.1 (page 51). In the next section, we will review some computational frameworks that make use of complexity theory concepts, illustrated by examples in the field of spatial epidemics.

1.5 Applications involving Spatial Epidemics

The applications of complexity theory in spatial epidemics follow basically three methodologies which can of course be combined: these are cellular automata, agent-based

[5] http://www.his.atr.jp/~ray/tierra/
[6] http://en.wikipedia.org/wiki/Artificial_life
[7] http://en.wikipedia.org/wiki/Complex_network

models and graph based (complex network) models. CA is a well established area in computation, and has been applied for many years in other areas which explains, at least partially, the great variety of CA based epidemic models, especially when compared to ABM. In fact, at the beginning of this study there were hardly any examples of ABM in spatial epidemics. Recently, however several models have been developed and some of them will be described in this section. Also, in the case of network models, there are hybrid models that include agent-based approaches.

A **Multi-agent system** can be defined as any computational system whose design is fundamentally composed of a collection of interacting parts (Reynolds, 1997). **Agent-based models**, also referred as Individual-based models, are a particular kind of multi-agent system where each "agent" corresponds to an autonomous individual in the simulated domain. A typical ABM consists of an environment (or framework) in which the interactions occur, and individuals are defined in terms of their behaviors (procedural rules) and characteristic parameters. The characteristics of each individual are tracked through time in contrast to modeling techniques where the characteristics of the population are averaged together and the model attempts to simulate changes in these averaged characteristics for the whole population (Reynolds, 1997). There is a lot of terminology to characterize agents in the field of AI, thus we will concentrate on a categorization described by Watt and Policarpo (2000). These authors distinguish between three types of agents:

- Reflex or reactive.
- Reflex or reactive with internal state.
- Goal-based.

Reflex or reactive agents are the simplest type of agent and they simply search in their database for conditions that match the perceived state of the world and perform the appropriate actions. These kinds of agents can be implemented using Finite State Machines (FSM) which are used to model the behavior of hardware (Watt and Policarpo, 2000). In a FSM, that we represent in figure 1-7, the machine is modeled as a memory

which samples the inputs and current state, and then decides what the next state should be.

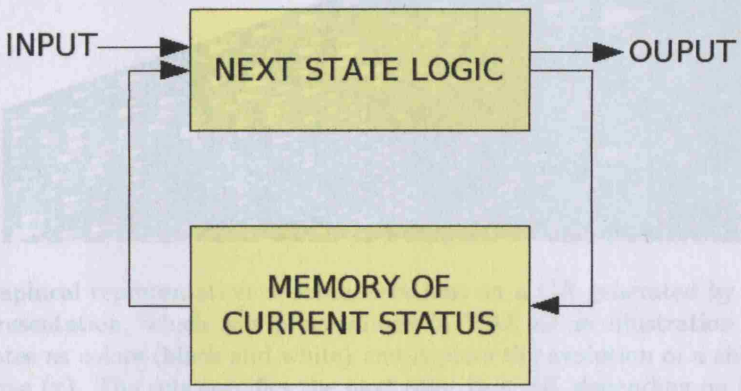


Figure 1-7: General model of an agent as a finite state machine. This diagram characterizes the behavior of the agent as an entity, that at any instant is in a certain state, and can only move to another state when a certain event, or group of events occur (based on Watt and Policarpo (2000).

Reflex agents with internal states differ from the previous ones because they have a memory of the past and thus can react differently to the same events over time. All reflex agents are entities that convert percepts into actions, with percepts being the unit of information supplied to the agent as an input. Goal-based agents, on the other hand, consider the consequences of their actions rather than simply reacting to their perception of the world. This is a more complex behavioral model from the cognitive point of view, since the agent has knowledge concerning its goals and how they can be achieved. Concerning space, ABMs can be spatially explicit and some spatially explicit ABMs also exhibit mobility. Spatially explicit models may use either continuous or discrete space. CA, which we show in figure 1-8, are a particular class of ABM characterized as being spatially-explicit, generally grid-based and generally containing immobile individuals who relate to the cells. They can be defined as discrete dynamical systems (in space and time) whose behavior is completely specified in terms of a local relation (Margolus, 1987).

Perhaps the most striking difference between CA and ABM is that one is based on a dense and uniform dissection of space, while the other is based on specific individuals

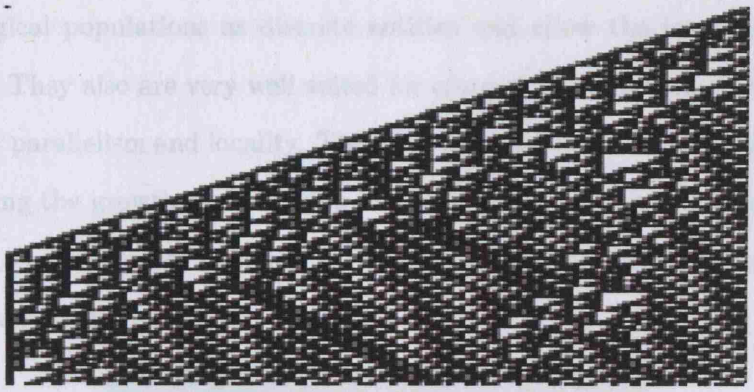


Figure 1-8: Graphical representation of some iterations on a CA generated by Wolfram's rule 110. This representation, which was programmed in VBA as an illustration for this thesis, encodes the states as colors (black and white) and it plots the evolution of a single row of cells (y axis) over time (x). The rule specifies the next color in a cell, depending on its color and its immediate neighbors.

distributed within space. This philosophical difference has a great impact on the simulations themselves and therefore the method should be chosen carefully, according to the phenomena under study. Figure 1-9 shows how MAS, ABM and CA relate to one another, with MAS containing ABM and thus containing CA. It is important to note here that although ABMs are more general, the first steps in ABMs were in fact through CA, with Von Neumann's universal copier and constructor (1940) and Conway's *Game of Life*(1970)[8].

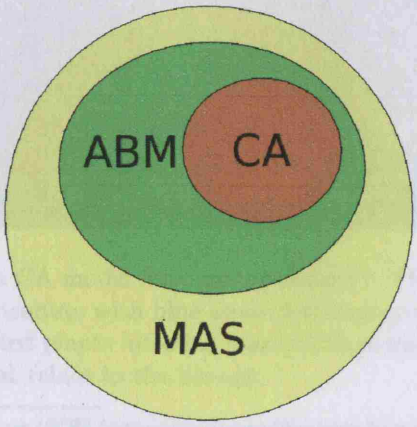


Figure 1-9: Relations between MAS, ABM and CA.

Some advantages of the use of CA for epidemic models are that they treat individ-

[8] http://en.wikipedia.org/wiki/Cellular_automaton

uals in biological populations as discrete entities and allow the introduction of local stochasticity. They also are very well suited for computer simulation, essentially due to their inherent parallelism and locality. There are some examples of CA epidemic models for representing the growth and diffusion of epidemics in plant populations. As plants cannot move, the pattern of infection is fully determined by local relations, as we can see in the example on figure 1-10.

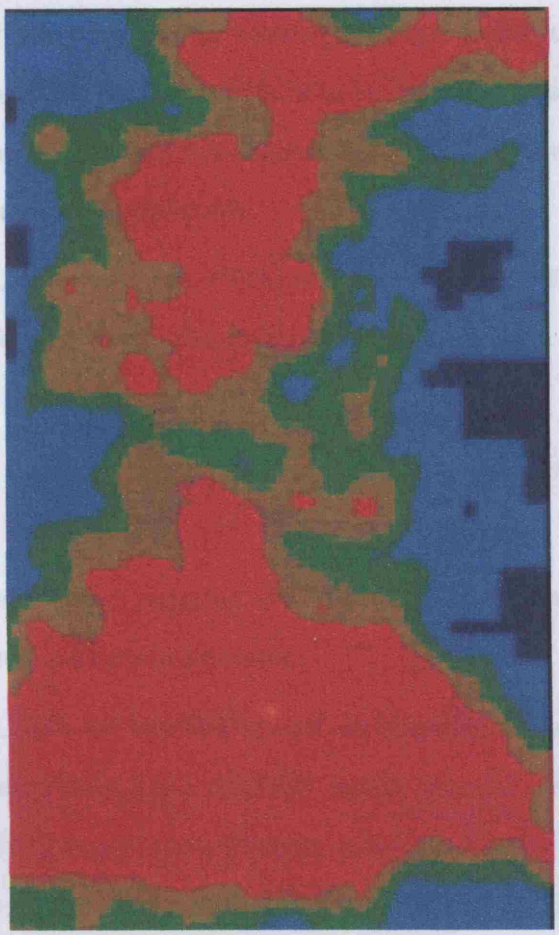


Figure 1-10: An example of a CA model for plant epidemics[9]. The map shows the distribution of the strawberry red core inoculum, with blue areas denoting greater disease incidence. In this case, disease spreads by infected plants infecting susceptible plants in their local neighborhoods according to specific rules that relate to the biology.

[9] http://www.scri.sari.ac.uk/SCRI/web/site/home/ResearchAreas/TopicsinPlantPathology/CerealPathologyatSCRI/modelling.asp

In human populations, CA models also have to deal with the problem of movement of individuals plus the interaction between individuals. There have been several approaches to this problem in recent studies (Fuentes and Kuperman, 1999; Conference of the First

Australian Conference on Artificial Life 2003, 2003; Fuks and Lawniczak, 2001; Mansilla and Gutierrez, 2000) which will be described in the next paragraphs.

Fuentes and Kuperman (1999) use an interaction radius to define the neighborhood and probability transition rules from one cell state to another; in this way they produce movement of individuals along the whole domain, with a certain associated probability. Conference of the First Australian Conference on Artificial Life 2003 (2003) also follow this radius approach; they use neighborhood radius to model contagion and motion radius to model movement, again with probabilities associated with each one. Figure 1-11 shows a simulation of this model where a cell represents an equal sized area of the landscape containing individuals.

Lattice Gas Cellular Automata (LCGA) is a particular class of CA where the dynamic transitions are performed in parallel over all the sites and can be decomposed in two stages:

- Propagation: the particles jump to a nearest-neighbor site according to their direction.
- Collision: the particles entering the same site at the same iteration, interact, producing a new particle distribution.

Some examples of LGCA are the HPP model, an acronym of the name of its inventors: Hardy, de Pazzis and Pomeau and the FHP model, this one invented by Frisch, Hasslacher and Pomea, both described in Chopard and Droz (1998). In their work, Fuks and Lawniczak (2001) use LGCA for reaction-diffusion systems applied to a generic disease. They considered a hexagonal grid with a maximum density of individuals in each cell and random movement into adjacent cells. This model allowed the authors to study the precise consequences of different strategies of vaccination and thus demonstrates an important application for this kind of models. Figure 1-12 shows two such simulations using two different spatial vaccination strategies: homogenic and ring vaccination. As we can see, the evolution of the pattern of infection is very distinct in both approaches which shows the importance of the strategy.

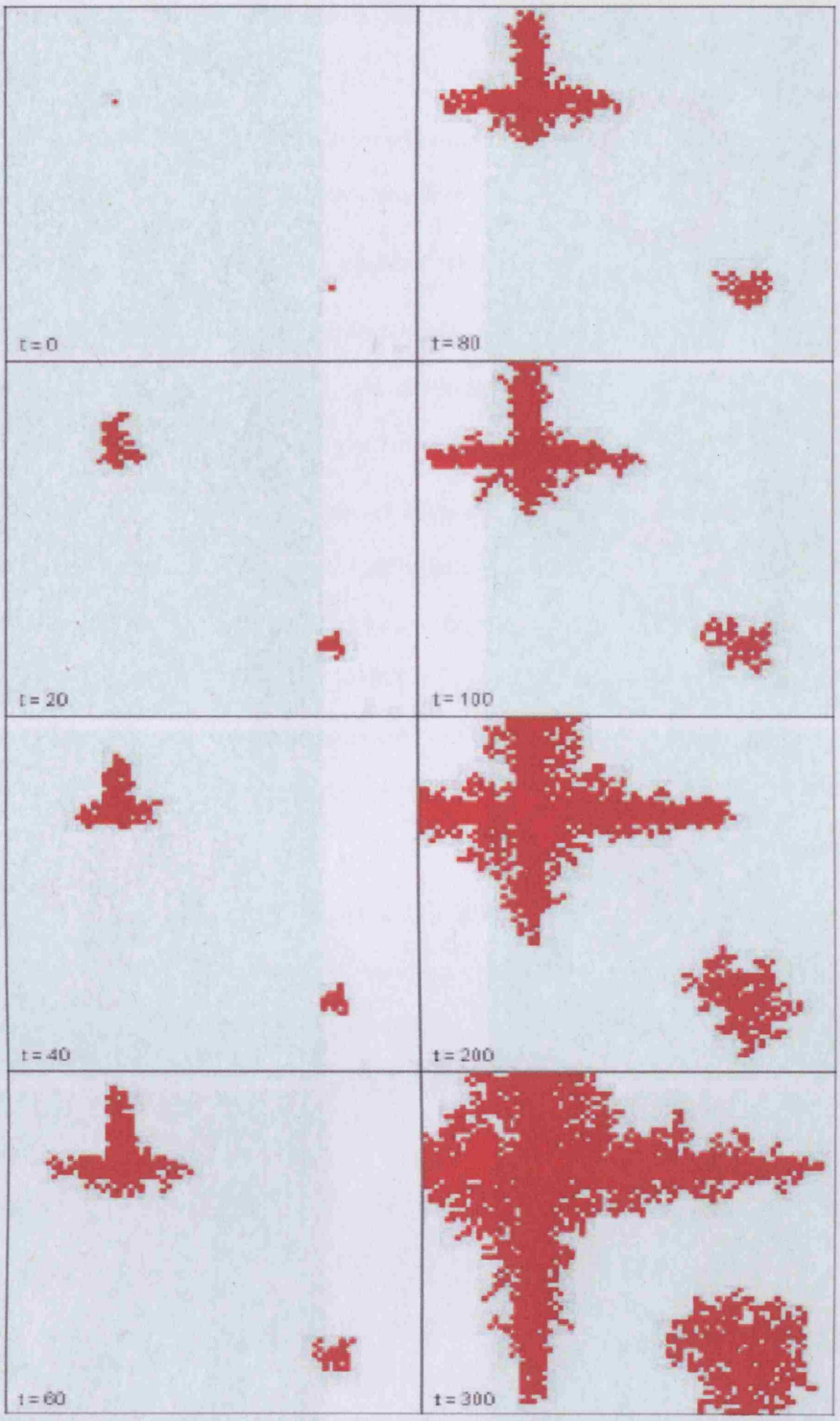


Figure 1-11: Map showing the status of the epidemic at t=0,20,40,60,80,100,200 and 300 in the Conference of the First Australian Conference on Artificial Life 2003 (2003) model. The shape of the outbreak is associated with a road link that allows the population to spread along it.

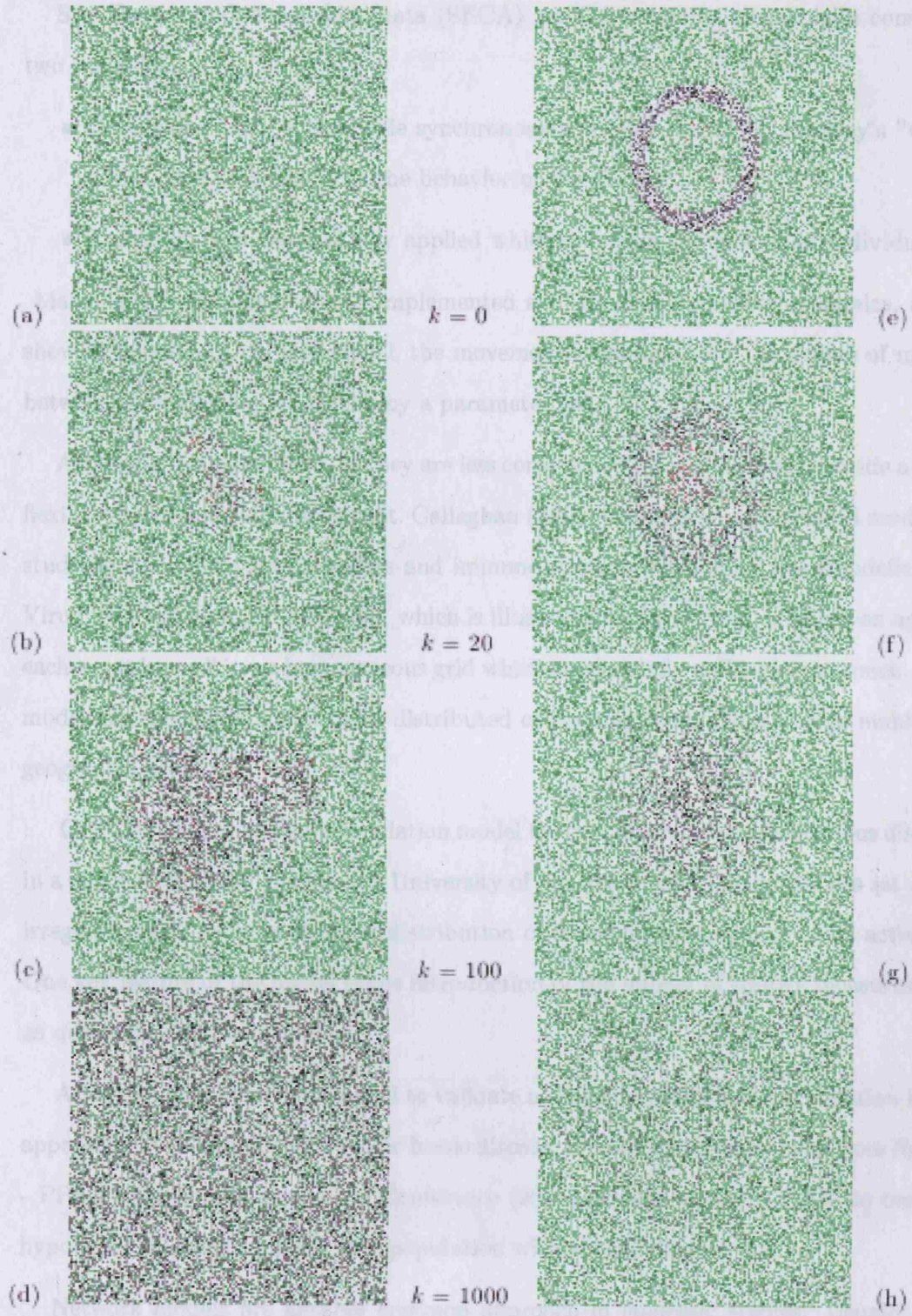


Figure 1-12: Fuks and Lawniczak (2001) LGCA simulation using homogenic vaccination (left) and barrier vaccination (right). Steps t=0,20,100 and 1000.

Site Exchange Cellular Automata (SECA) are CA networks whose rules consist of two sub rules:

- Contagious rule - a local rule synchronously applied, based on Conway's "Game of life" and thus describes the behavior of the disease.
- Transport rule - sequentially applied which describes the motion of individuals.

Mansilla and Gutierrez (2000) implemented a SECA for modelling epidemics, as we show in figure 1-13. In this model, the movement is random, but the degree of mixing between individuals is controlled by a parameter (p).

ABMs, on the other hand, as they are less constrained by spatial rules, provide a more flexible way to deal with movement. Callaghan (2005) presents an agent-based model for studying the population dynamics and immune response to Human Immunodeficiency Virus (HIV). Although the model, which is illustrated in figure 1-14, is based on agents, each agent is a cell in an homogeneous grid which is essentially a *CA-like* approach. This model has the advantage of using distributed computing to simulate a large number of geographic areas.

CUPUM (2005) present a simulation model for the transmission of infectious diseases in a small geographic space in the University of Southampton. The model has set up an irregular space, with an unequal distribution of individuals according to its activities. One key feature of the model is the introduction of the impact of control measures such as quarantine and vaccination.

ABM can also be used as a tool to validate a standard mathematical equation based approach. In their study of a vector borne disease, Parallel Problem Solving from Nature - PPSN VIII: 8th International Conference (2004) develop a simple ABM to test the hypothesis of perfect mixing in a population with spatial dependence.

Network models are another common approach in epidemic studies. Usually they rely on social networks and since the spread of a disease also occurs by person-to-person contact, the structure of the network of such contacts has a huge impact on the nature of the epidemic. Small-World networks are particularly popular for representing all kinds

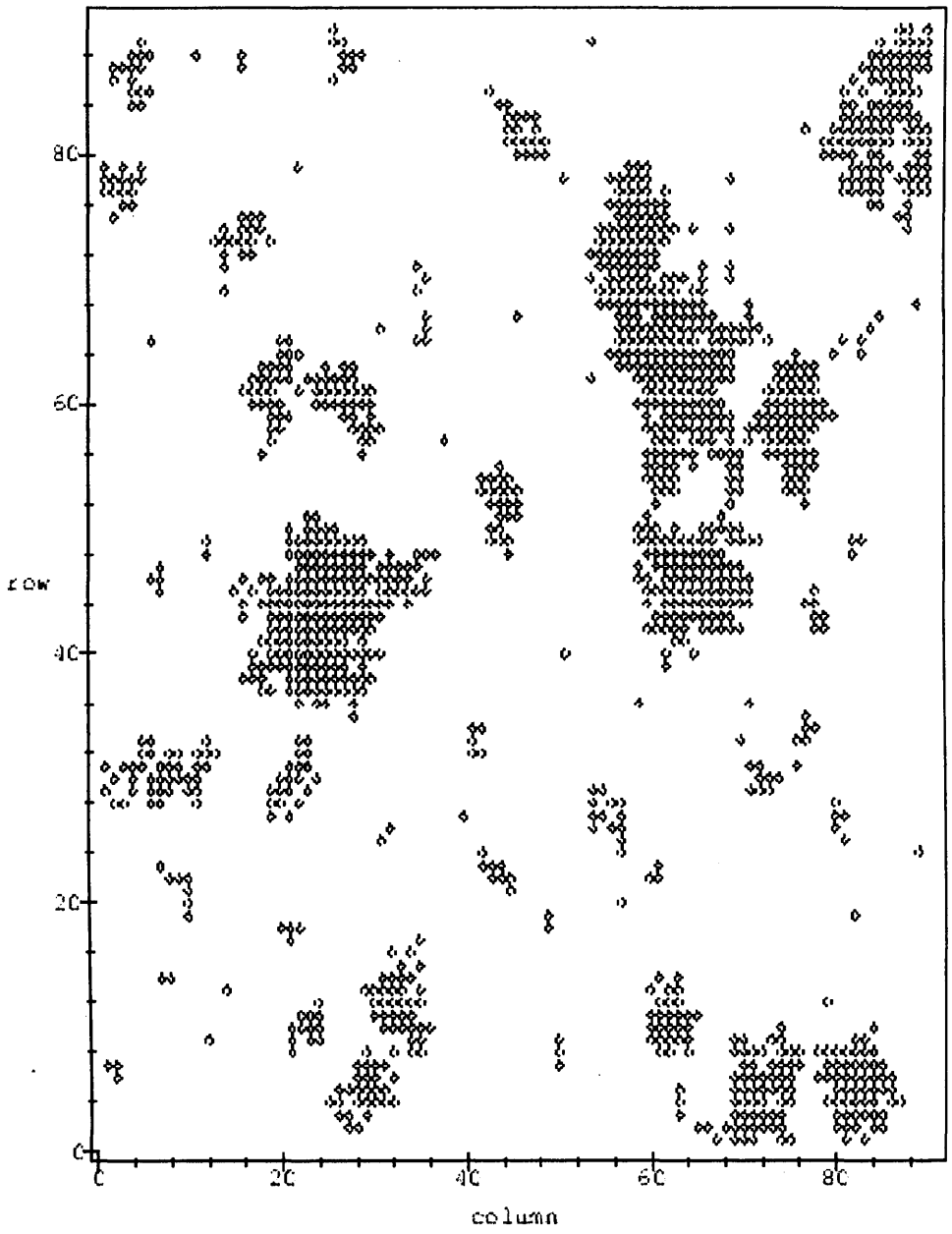


Figure 1-13: The configuration generated by a schematic contagious rule on Mansilla and Gutierrez (2000) SECA epidemic model. The infected sites are represented in dark color;

of social networks and thus it does not come as a surprise that they are the basis for many epidemic models. Newman (1999) points out that disease models which incorporate a measure of susceptibility to infection have a percolation transition at which an epidemic sets in whose position is influenced strongly by the SW nature of the network. Since

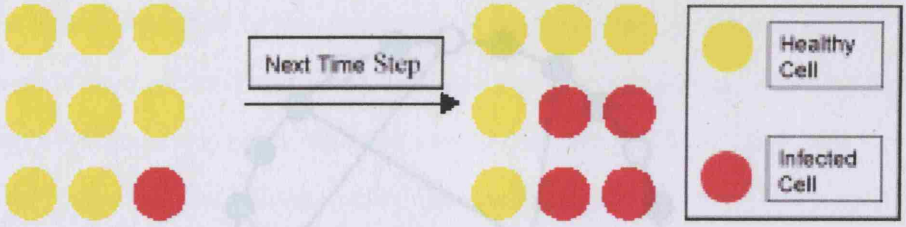


Figure 1-14: Illustration of a rule applied by Callaghan (2005), where a cell becomes infected at time step t+1 if at least one of its neighbors was infected at time step t.

this kind of network was actually considered in the conceptual model, we will develop this topic more in section 2.2.1 (page 51). In the next paragraphs we will only refer to some SW epidemic models found in the literature in a summary manner.

Moore and Newman (2000) developed a site and bond percolation model for diseases, which is also a probabilistic CA model. The occupation probabilities of sites correspond to the susceptibility of individuals to the disease and the occupation probabilities of bonds correspond to the transmissibility of the disease. The model is built on a SW network as we show in figure 1-15.

Chen (2001) developed a model for Foot and Mouth Disease that combines two approaches: compartmental models for individuals moving inside the farms, and lattice based methods for agents moving between farms (using SW). This model considers two different scales for virus transmission routes: local (neighbor farms) - that we see in figure 1-16 - and long-distance (countrywide movement). A variable P (0 to 1) controls the degree of disorder of the network, from structured neighborhood movement to totally random movement.

Finally, it is important to note an inspirational work from Goteborg University and Chalmers[10], which is a CA model for the propagation of diseases around the world. In this model, each cell represents the population living in a certain area. It is assumed that people living in highly populated areas are more likely to interact with more people and that people living in parts of the world with more developed infrastructure tend to travel larger distances, which explains the structured diffusion. The infection rule applied to each cell is expressed by (1.4). For the W parameter, several traveling possibilities are

[10] http://www.dd.chalmers.se/~f97kahe/epidemicsim/epidemicsim.htm

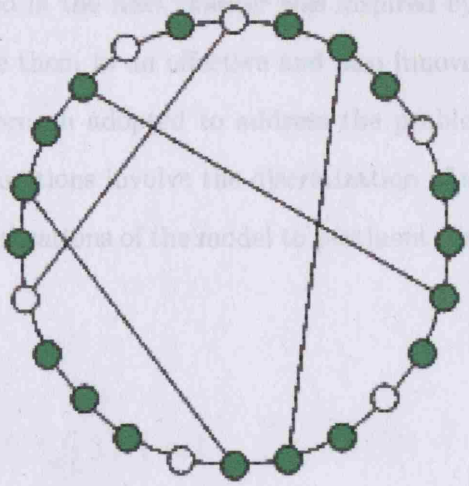
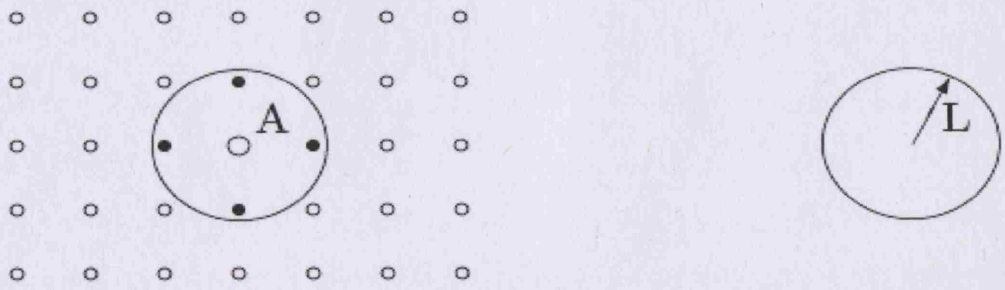


Figure 1-15: SW model from Moore and Newman (2000) that starts from a regular lattice and adds a number of random shortcuts.



. Figure 1-16: The definition of local neighbors of a farm A using radius L.

considered, including the SW component.

$$\frac{I_c ont S W_{1x,y} W_{2x} W_{2y} N_0 C_0}{area_c ell} \tag{1.4}$$

$I_c ont$ is the number of people that are contagious in the source-cell X, S the number of people that are susceptible in the target-cell y, $W_{1x,y}$ a weight between 0 and 1, which is proportional to the fraction of time a person living in the target-cell x in average spends in the source-cell Y during a time step, W_{2x} and W_{2y} are weights depending on the infrastructure in the source and target-cell respectively, N_0 and C_0 are the number of interactions and contagiousness and $area_c ell$ the area of the cell in Km^2.

The model described in the next chapter was inspired by some of these studies and it attempts to combine them in an effective and also innovative way. A major issue is the computational approach adopted to address the problem, based on CA and SW, but other important questions involve the discretization of space, the representation of movement, and the applications of the model to pertinent questions such as vaccination.

Chapter 2

The Conceptual Model

2.1 Objectives of the Model

This model attempts to simulate the **spreading of a non vector disease**, a disease that spreads directly from one host to another without the intervention of an organism, such as a mosquito, for carrying the disease-causing microorganisms. The model simulates the evolution of the disease in space and time, **in a human population with spatial dependence**, which brings emphasis to its spatial component and therefore the use of GIS. As noted in the literature review in section 1.5 (page 36), movement of the population is an important process in this kind of model as much as the infection process itself. A lot of the work reviewed in this field (for instance: Mansilla and Gutierrez (2000) and Fuks and Lawniczak (2001)) follows the assumption of random mixing of the population: individuals moving randomly from one place to the other. This is a way of overcoming the lack of data or lack of knowledge of the mechanisms of movement. However if it is a weak assumption for some animal populations (like ants), it is certainly unrealistic for humans, as we know they follow structured patterns in their periodic movements which are unlikely to be random. One of the main concerns of this model was to put aside this simplistic assumption and develop a realistic movement model. It was also intended to build a realistic infection model considering latent and morbid periods, in order to provide it with flexibility to simulate different kinds of diseases.

In the next section, the movement and infection models will be described in more detail, and then we will explain how they were combined using the ALife paradigm. There were two implementations of this model: the first one using CA, and the second one, which was adopted as the main one throughout the rest of this thesis, using agents moving in a continuous space (an ABM).

2.2 Description of the Model

2.2.1 The Movement Model

A graph is a pair $G = (V, E)$ where V is a set of vertices and E is a binary relation on V. E contains a pair (a, b) if there is an edge between the a and b vertices. If the graph is directed, this pair is ordered: thus (a, b) is not the same as (b, a). Figure 2-1 shows an example of this kind of graph.

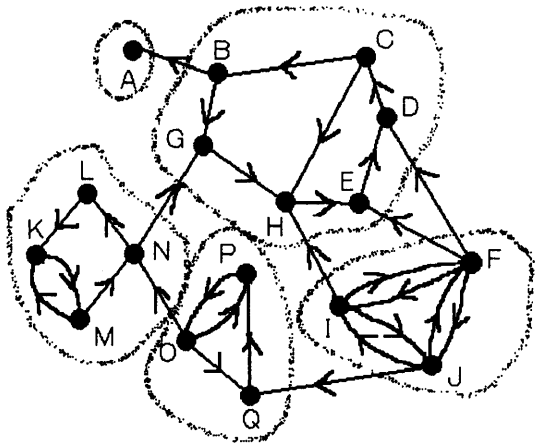


Figure 2-1: Example of a directed graph: in this graph, the pair (A, B) is not the same as the pair (B, A)[11]. In a directed graph, also called *digraph*, each edge is given a unique direction (Weisstein, 2007).

[11] `http://www1.odn.ne.jp/~cei82850/graph01j.gif`

In social networks, the system is represented by a graph where the vertices are members of a population and the edges are ties between the members (Zheng et al., 2003). This implies that adjacent vertices share the same kind of social relation such as being neighbors, family, friends or partners. This kind of graph is simple and undirected; simple because there are no loops (an individual cannot be connected to itself) and there are no duplicate edges (a pair of individuals cannot be friends *twice*!); undirected because the relation is the same from individual a to b as from individual b to a, as we show in figure 2-2. The problem of understanding an epidemic is closely related to what kind of social network is involved [12].

[12] `http://www.ams.org/featurecolumn/archive/networks4.html`

The SARS explosion in 2002 was complicated by the network of global mobility which has expanded enormously during the second half of this century. This is a network representing casual co-presence, rather than a social network like the one underlying AIDS, which requires more intimate ties between individuals. The differences in the topological structure of these networks may explain the different degrees of contagiousness of SARS and AIDS. In this model it was simulated a contagious disease that depends only on physical proximity. Therefore, the contact model is supported not by the acquaintances network as in so many epidemic models, but by a movement model that is based on geographical space rather than "social" space. Due to the non-homogeneous arrangement of space with irregular neighborhoods and an irregular distribution of individuals, there are some nodes with more contacts than others. This connectivity does not have any meaning in social terms for as we have said, the contact process does not require any ties between individuals. However, it represents the fact that due to the place where they live, some individuals are more exposed to infectious contacts than others. This is an interesting feature to analyze, and thus we will study this kind of connectivity of individuals in section 4.2 (page 117) despite its lack of social meaning.

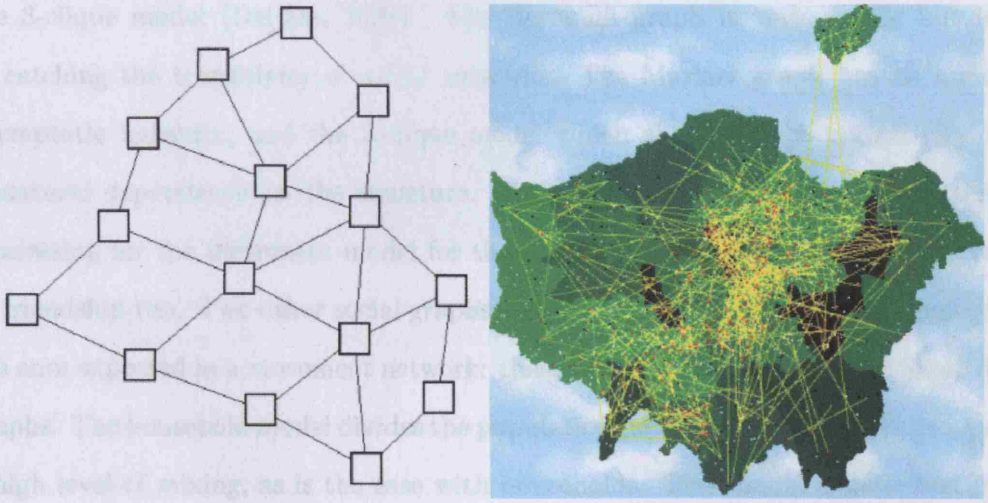


Figure 2-2: Undirected graphs: a) general example[13] and b) a graph generated in this model for illustrative purposes only, recreating the movement of people in Greater London.

[13] http://www.aisee.com/png/undir.htm

This study is also concerned with relaxing the assumption of homogeneous mixing

which is a fairly unrealistic approach in human populations. There are not many studies focused on movement networks but there are some studies on social networks. To represent social networks, it is important to focus on three properties: the bounded neighborhood size, high transitivity and realistic dependence from the structure (Deijfen, 2000). The first property is related to the fact that our circle of friends is finite and so there should be a limited number of neighbors for each node. The second property reproduces the fact that we expect most of our friends to be friends of each other, and so the probability that a certain edge is present given the absence of presence of all other edges is larger if its end vertices are second neighbors than if they are not. Finally, the third condition makes sure that there are no unnatural dependences from the structure among the edges; that is, the absence or presence of a certain edge should not depend on information regarding parts of the graph that are located far away from the edge. This is because we assume that individual behavior can be influenced by friends, but it is less likely that to be influenced by people far away in the social network which it does not have any kind of relation to.

Some examples of random graphs are the Bernoulli graph, the Markov graph and the 3-clique model (Deijfen, 2000). The Bernoulli graph is very simple but it fails in catching the transitivity of social networks. The Markov graph has an unnatural asymptotic behavior, and the 3-clique model which shows a high transitivity, has a unnatural dependence on the structure. However, these graphs are not particularly interesting for the movement model for the properties they try to emulate are related to friendship ties. Two other social graphs show some features that can be analogous to the ones expected in a movement network: these are the household and the Small-World graphs. The household model divides the population into mutually exclusive groups with a high level of mixing, as is the case with households. This model is described by two separate graphs: one describing the division into households and another one describing the relationships between individuals. Although the household graph itself does not allow the spreading of the disease, households are linked together by the friendship network. The movement model developed in this study establishes an analogy between

households in social space and regions in geographical space. Given a set of N labeled vertices, let ς_1 be a graph representing the household structure and ς_2 a random graph representing the network of contacts. The network ς is formed by the super imposition of the graphs ς_1 and ς_2 as in (2.1).

$$\varsigma = \varsigma_1 \cap \varsigma_2 \tag{2.1}$$

In the model developed in this thesis, instead of households there is a set of geographical regions such as the ones shown in figure 2-3.

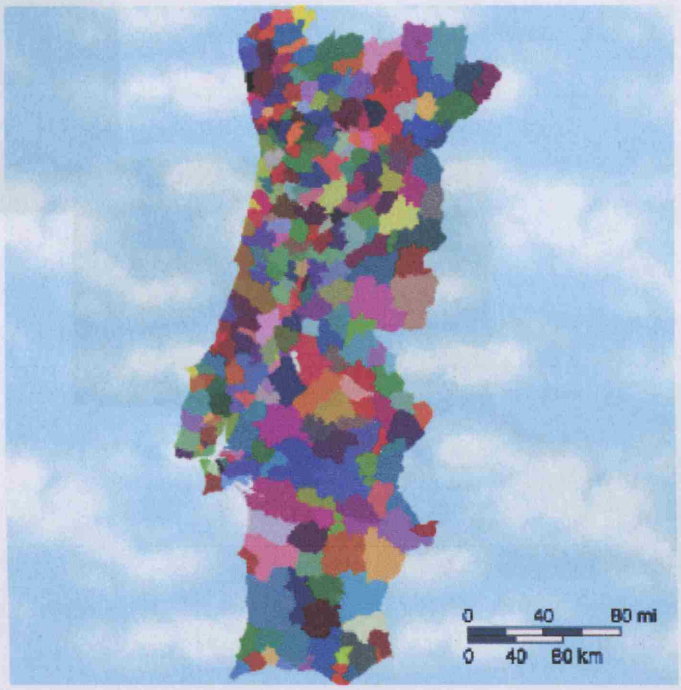


Figure 2-3: ς_1 based on the Portuguese regions *Concelhos* described in section 5.1 (page 161). In this map we see *Concelhos* represented with distinct colors.

In this sense, these regions are taken as having a high degree of mixing although, as we will see later, they are not necessarily homogeneous. This is true if we think about cities where the geographical aggregation imposes some kind of identity that assembles individuals into a common domain. The movement model establishes a network linking neighboring cities (adjacent vertices) and another network inside the city itself; the difference from the household model is that the network inside the city is not a random

graph. Each city is subdivided into smaller units called neighborhoods. A random graph defines the movement inside each neighborhood and another random graph links these neighborhoods together inside the city. The choice of random functions for governing these graphs is essentially due to the lack of data/knowledge about their structure and also to the purpose of introducing some variability in the model. To summarize, we can say that there are three nested levels (or scales) involved in this movement model: the global scale, regions (split into two levels), and neighborhoods as we illustrate schematically in figure 2-4.

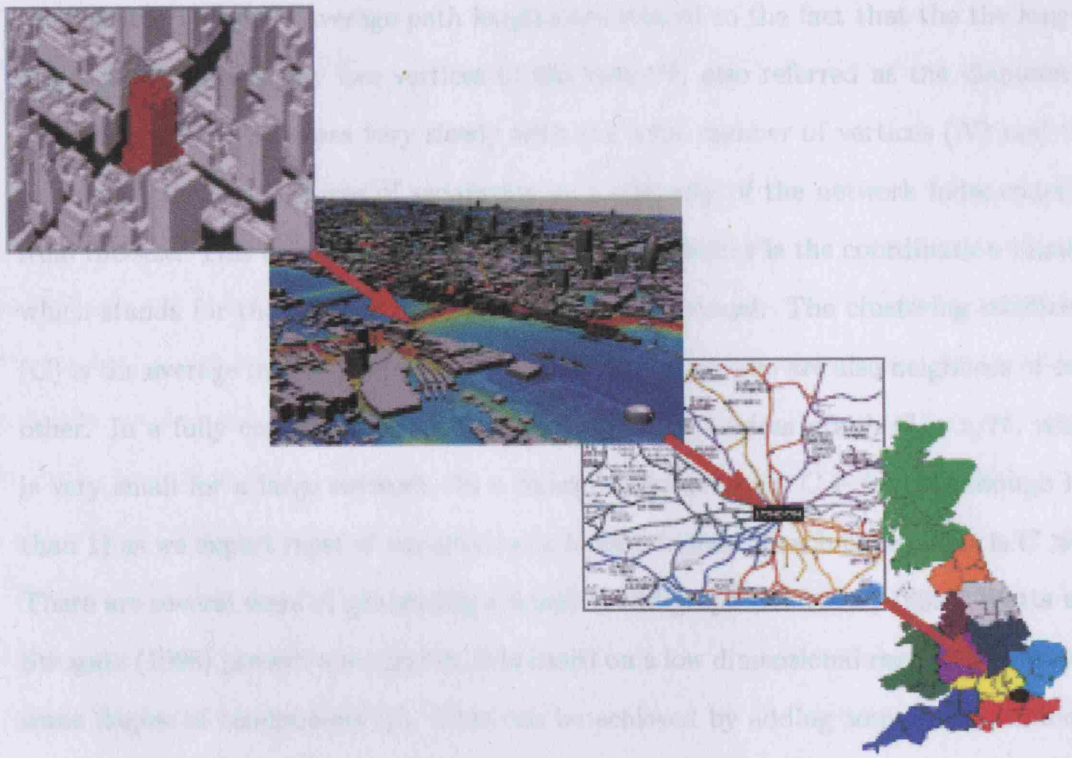


Figure 2-4: Different scales involved in the movement model. From top to bottom: neighborhood, intra-region, inter-region and global.

Elaborating more on each kind of movement, we can say that neighborhood movement (ς_1) represents the movement of a individual in a street or a block, involving activities such as staying at home and going to the pub. Intra region movement (ς_2) is the movement inside the city, for instance going to work or shopping while inter region movement (ς_3) is traveling to neighboring cities, for instance visiting relatives, or for taking part in leisure activities. Finally, random movement (ς_4) links together distant

parts of the network and thus is related to another kind of social graph: the Small-World network.

Small-Worlds, also referred popularly as "six degrees of separation" were first introduced by Milgram in 1967, and are now widely used to represent several kinds of social networks (Newman, 1999). The name is associated with the fact that any two people chosen randomly on the planet, would be connected to each other by an average chain of six degrees of acquaintances (Newman, 1999). Small-worlds can be defined as random graphs which possess two properties: short average path lengths and high clustering coefficients. The short average path lengths are related to the fact that the the longest direct path between any two vertices in the network, also referred as the diameter of the network (D), increases very slowly with the total number of vertices (N) and this establishes the short degree of separation as a property of the network independently from the size. This is expressed on $D = logN/logz$, where z is the coordination number which stands for the number of connections per individual. The clustering coefficient (C) is the average fraction of pairs of neighbors of a node who are also neighbors of each other. In a fully connected graph, $C = 1$, while in a random graph $C = z/N$, which is very small for a large network. In a Small-World network, C is high (although less than 1) as we expect most of our friends to be also friends of each other, that is $C \gg 0$. There are several ways of generating a Small-World graph (Newman, 1999). Watts and Strogatz (1998) present a model which is based on a low dimensional regular lattice with some degree of randomness (p). This can be achieved by adding some links randomly which will produce the shortcuts responsible for a short average path length. Figure 2-5a shows a regular lattice where a small number of random links had been added.

In figure 2-5b another Small-Worlds model (Kasturirangan, 1990) is presented where a few nodes have high coordination numbers. In fact, this model presents another property which exists in some Small-World models: that is the power law distribution of the connected nodes. The probability $P(k)$ that a node will have connections to k other nodes is given by $P_k \sim k^{-\gamma}$, which suggests that a small number of individuals have a high number of acquaintances. Although at first sight these properties only

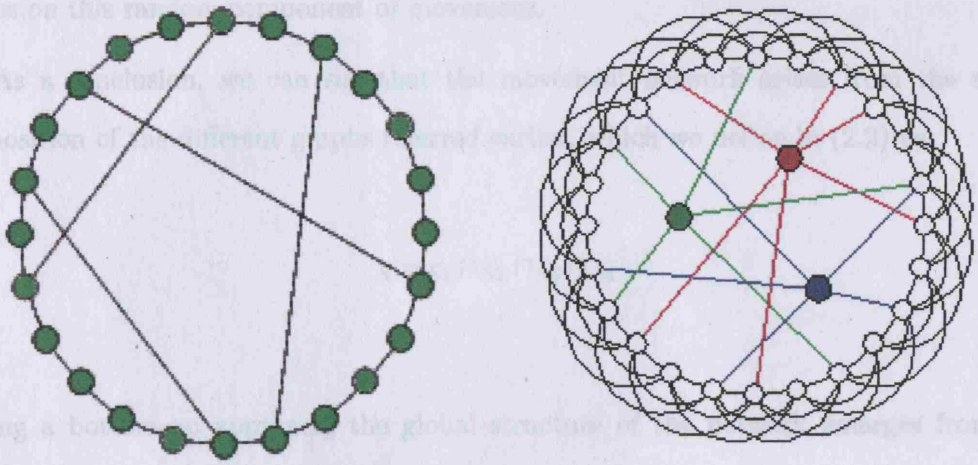


Figure 2-5: Small-World models: a) from Watts and Strogatz (1998) and b) from Kasturirangan (1990), as presented in Newman (1999).

describe ties between individuals, they are of some interest for the movement model. For studying movement, it is important to consider two different networks based on individuals and their accessibilities to each other. The individuals network is related to their activities but it is influenced by the accessibility network. In this work, we consider only the network of individuals but accessibilities are implicit. The movement network is constructed over time, where each node represents the location of an individual a at time step t and edges connect locations at time step t and time step $t-1$. Following the Watts and Strogatz model (Watts and Strogatz, 1998) we introduced small values of p on ς_4, that produce long range links in the network. In terms of mobility, this can be explained as a small number of individuals that travel large distances over the domain, as is the case of politicians, football players, or businessmen. This assumption for the movement network is analogous to the assumption of very well connected individuals in the Kasturingaran model (Kasturirangan, 1990). On another level, regions with higher populations, for example in big cities, show high coordination numbers, behaving as hubs. It should be noticed that this assumption follows the idea that the accessibilities follow the population distribution, and thus can compensate for the lack of accessibility data. The individuals network is also highly clustered inside each region, reproducing in this way another Small-World property. In section 4.2 (page 117), some experiments

focus on this random component of movement.

As a conclusion, we can say that the movement network arises from the superimposition of the different graphs referred earlier, which we define in (2.2) as:

$$\varsigma = \varsigma_1 \cap \varsigma_2 \cap \varsigma_3 \cap \varsigma_4 \tag{2.2}$$

.

Using a bottom-up approach, the global structure of the network emerges from the displacement of each individual a, according to the equation 2.3.

$$a_{(i,j)}^{(t+1)} = a_{(i,j)}^{(t)} + d \tag{2.3}$$

The stochastic variable d has a probabilistic distribution, according to equation 2.4.

$$d = (P_1)D_1 + (P_2)D_2 + (P_3)D_3 + (P_4)D_4 = \sum_{x=1}^{4}(P_x)D_x = 1 \tag{2.4}$$

The parameters D1, D2, D3 and D4 are the different ranges of movement: neighborhood, intra-region, inter-region and random. Figure 2-6 shows the proportions suggested for each component of movement. Following a distance decay law, we assume an individual is more likely to stay where it is than travel over large distances. We are aware that the probability of travelling is not homogeneous, once it may depend for instance in the socio-economic status, but this rule tried to reproduce in a simple way what happens to the most part of the population without bringing other sources of variation to the model, although often this kind of simplifications may lead to biased scenarios. Movements with a longer range (such as inter-region and random) were given a lower probability than movements with a shorter range (such as intra-region and neighborhood).

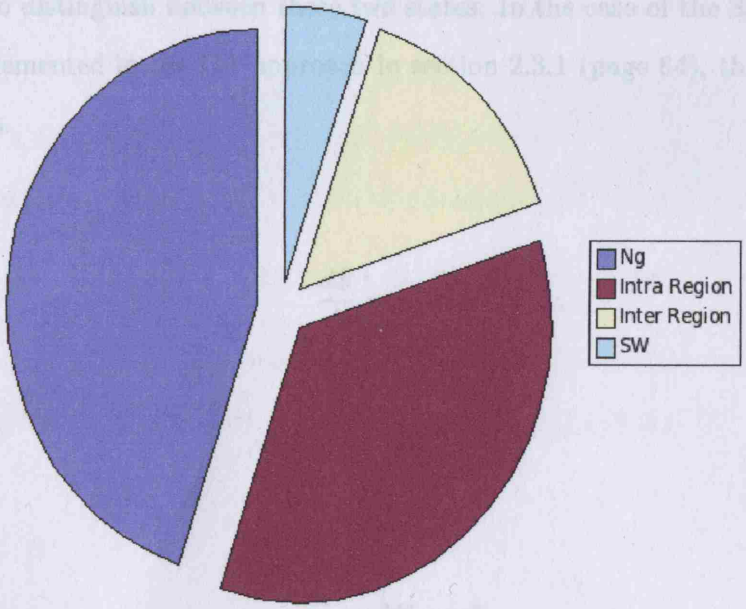


Figure 2-6: The distribution of probabilities for the different movement components adopted throughout this thesis. These are empirical values based on a general distance decay law: shorter range movements have a higher probability to occur than larger range movements; the random component, which is associated to Small-Worlds theory, has a very small probability.

2.2.2 The Infection Model

The SIR model (Susceptible-Infectious-Removed) formulated by Kermack and McKendrick in 1927 Epstein (1997) expresses the relations between the different population states, in terms of differential equations. These states are:

- Susceptibles (S) - individuals capable of acquiring the disease.
- Infected (I) - individuals who can transmit the disease.
- Removed (R) - individuals who are either dead, recover or become immune from the disease.

There are several variants of this model: SIS (susceptible-infected-susceptible), SIRS (susceptible-infected-removed-susceptible) and SI (susceptible-infected) are examples. In the ABM developed in this thesis, the SEIR (Susceptible-Exposed-Infectious-Removed) in equations (2.5), (2.6), (2.7) and (2.8) was developed which introduces a fourth state very relevant in childhood diseases such as mumps or measles: the Latent or Exposed (E). In this model, an **infected** individual might not be **infectious** and therefore it is

important to distinguish between these two states. In the case of the SIR model which will be implemented in the CA approach in section 2.3.1 (page 64), the two terms are synonymous.

$$\frac{dS}{dt} = -\beta SI \tag{2.5}$$

$$\frac{dE}{dt} = \beta SI - \theta E \tag{2.6}$$

$$\frac{dI}{dt} = \theta E - \alpha I \tag{2.7}$$

$$\frac{dR}{dt} = \alpha I \tag{2.8}$$

Latent or Exposed are individuals that acquire the disease but are still not infectious. In this model, contrary to what happens in the SEIRS model, there is no possibility of re-infection after recovering from the disease; this means the removed individual cannot become susceptible again. It is important to emphasize that there is no distinction between dead and recovered but as the disease simulated by this model is rarely lethal, it is reasonable to assume that the removed individuals become immune rather than dead. The contact parameter β, also known as infection force, regulates the transition from susceptible to latent. The transition parameters θ and α regulate the transition from exposed to infectious and infectious to removed. From the point at which the individual gets infected until it becomes infectious, there is a latent period (that includes the symptomatic and assymptomatic period), and from the point that it gets infectious until it is removed, there is an infectious period that will be referred henceforth as the morbid period. Figure 2-7 shows the sequence of state transitions and how the latent and morbid period are defined.

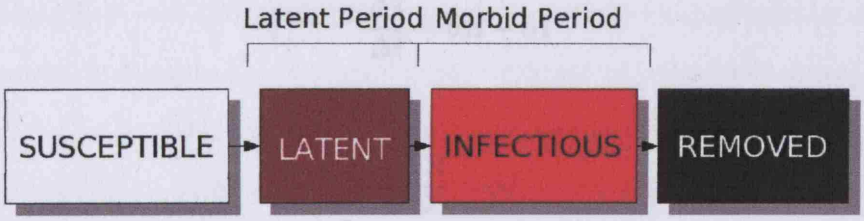


Figure 2-7: The epidemic state changes through time.

One fundamental quantity governing the time evolution of these equations is the epidemic threshold or growth rate: R_0, which is defined in (2.9) as:

$$R_0 = \frac{\beta S}{\gamma} \tag{2.9}$$

This ratio expresses the expected number of secondary cases per primary case in the population and it establishes the turning point at which the epidemic occurs. If $R_0 < 1$ the epidemic disappears whereas if $R_0 = 1$, it becomes endemic in the population. If $R_0 > 1$ there is a asymptotic probability of a large outbreak of the disease.

In more complicated models, R_0 may have to be expressed in a different way, for instance depending on the population composition and level of sexual activity (Diekmann and Heesterbeek, 2000) but it can always be defined as a non negative function of the model parameters, satisfying the conditions expressed above (Deijfen, 2000).

The discrete version of this model is given in equations (2.10), (2.11), (2.12) and (2.13).

$$\frac{\Delta S}{\Delta t} = -\beta SI \tag{2.10}$$

$$\frac{\Delta E}{\Delta t} = \beta SI - \theta E \tag{2.11}$$

$$\frac{\Delta I}{\Delta t} = \theta E - \alpha I \tag{2.12}$$

$$\frac{\Delta R}{\Delta t} = \alpha I \tag{2.13}$$

Where the Δ symbol represents a discrete change over a time period t to $t+1$. Therefore the variables at $(t+1)$ can be computed as in equations (2.14), (2.15), (2.16) and (2.17) as:

$$S^{t+1} = -S^t(\beta I^t - 1) \tag{2.14}$$

$$E^{t+1} = \beta S^t I^t - E^t(\theta - 1) \tag{2.15}$$

$$I^{t+1} = \theta E^t - I^t(\alpha - 1) \tag{2.16}$$

$$R^{t+1} = \alpha I^t - R^t \tag{2.17}$$

The final size of the epidemic is the fraction of the population that becomes infected, sooner or later. In this study, a closed population of N individuals was assumed as we show in equation (2.18).

$$\frac{dN}{dt} = 0 \tag{2.18}$$

As the time scale of the disease under study is quite short, it is plausible to ignore the demographic turnover (Diekmann and Heesterbeek, 2000)

We also considered that we are dealing with micro parasites, cases in which a single infection triggers an autonomous process in the host (Diekmann and Heesterbeek, 2000).

This means that each process is independent from the other individuals and from the environment, a situation that does not occur with macro parasites. As we explained in section 2.2.1 (page 51), in this model the contact process is only a function of physical proximity and is not based on social ties as for instance is the case of sexually transmitted diseases. Therefore the contact is directly determined by the movement model: to occur infection it is only necessary to have immediate proximity between a susceptible and an infective and we called this distance parameter the infection radius.

Finally, it is important to enforce the difference between the model adopted in this study and the compartmental model formulated by Kermack and McKendrick (Epstein, 1997). This difference relies on relaxing the assumption of homogeneous infectivity and homogeneous mixing of susceptibles and infectives. Rather than such an assumption, it considered complex spatial scenarios with irregular distributions of individuals, and thus this model will be considered to be more realistic. As the homogeneous assumptions were abandoned, the final size as well as the growth rate of the epidemic can no longer be computed analytically and these can only be achieved through simulation (see section 5.2, page 167).

2.3 The Computational Paradigm

As already referred to section 1.5 (page 36), the method chosen to implement the model has important implications for the way the phenomena is seen. In ALife, the behavior of the model is as important or even more important than the results, for the focus is on simulating the ongoing dynamics, rather than achieving a final state. Similar to what happens in a real laboratory, the method chosen for the simulation is crucial, with the difference in this case that the laboratory is the computer.

The stages of a generic modelling process that were the stages followed in this study, are represented in figure 2-8. The process starts with the formulation of a model of reality, followed by the formulation of its assumptions. The next step is to transform it into a computer model and to run it. The interpretation of the results leads to the

validation of the model. If the model is validated it can be used for explaining reality and forecasting new scenarios. If not, then the initial assumptions may have to be changed.

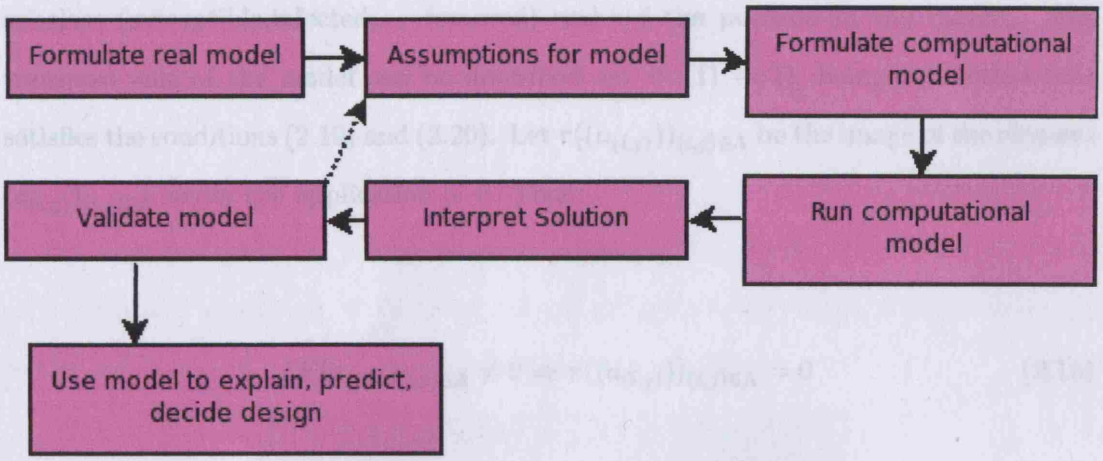


Figure 2-8: Stages of a generic modelling process.

2.3.1 The CA Approach

The first implementation of this model followed a SECA approach which was already described in section 1.5 (page 36). This is a typical CA model where each time step is split into two phases: mixing (or movement) and infection. Each of these phases, shown in figure 2-9, has its own rules, determined respectively by the movement and infection models.

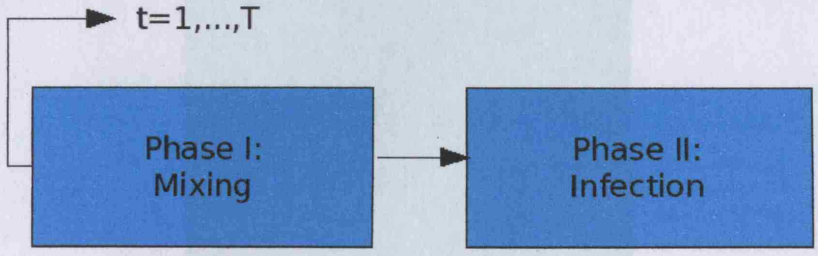


Figure 2-9: The SECA model: each time step is subdivided in two sequential phases: mixing and infection. These correspond to two subsets of rules.

Let Z be the set of integer numbers and $\Lambda \subseteq Z^2$ be the lattice. Λ represents the space where the population lives. At each time step t, a site a of Λ is either empty

or occupied, thus each site represents an individual. Occupied sites move according to transport rules. Let Ω be the configuration space of the matrix. $\Omega = 0, 1, ..., p^{\Lambda}$ is the set of elements on the form $(a_{(i,j)})_{(i,j)\in\Lambda}$, where $a \in 0, 1, ..., p$ represents their subclass (susceptible,infected,..., removed) and i, j the position in the matrix. The transport rule of the model can be described as: $\tau : \Omega \rightarrow \Omega$, being a function that satisfies the conditions (2.19) and (2.20). Let $\tau((a_{(i,j)}))_{(i,j)\in\Lambda}$ be the image of the element $(a_{(i,j)})_{(i,j)\in\Lambda}$ under the application of τ. Then:

$$IF(a_{(i,j)})_{(i,j)\in\Lambda} \neq 0 \Rightarrow \tau((a_{(i,j)}))_{(i,j)\in\Lambda} = 0 \tag{2.19}$$

For every $x_1, x_2 \in \Lambda$, where $x_1 \neq x_2$:

$$IFa_{x1}, a_{x2} \neq 0 \Rightarrow \tau(a_{x1}) \neq \tau(a_{x2}) \tag{2.20}$$

In other words, every nonempty element of the matrix can only move to an empty site and two occupied sites cannot move to the same empty site. Let us assume that the lattice contains a subset of units called σ and that $\Lambda \subseteq \sigma$ as we see in figure 2-10.

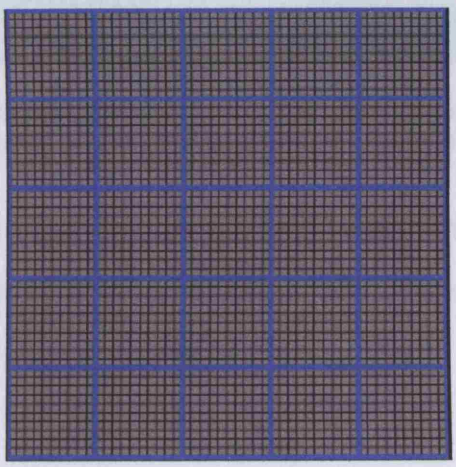


Figure 2-10: Lattice Λ, with units σ (blue) and sites (gray);

The rules applied to every $(a_{(i,j)})_{(i,j)\in\Lambda}$ are probabilistic and consider that the movement can be decomposed into the four states, described in section 2.2.1 (page 51):

neighborhood movement, intra-σ movement, inter-σ movement and random movement. The probabilities assigned to each kind of movement are represented in figure 2-6. With neighborhood movement, each cell can only move to one of its eight adjacent neighbors (the Moore Neighborhood) which is chosen randomly while with intra-σ movement, the cell can occupy any position randomly chosen inside σ. With inter-σ movement, the cell can only move to one of the adjacent σ (σ Moore neighborhood) and occupy a random position there. Finally, for random movement the cell can move to anywhere in the domain: this is the movement type related to SW theory. Figure 2-11 shows a screenshot of a simulation, considering only neighborhood movement. For the infection model, a SIR model with Moore neighborhood was adopted.

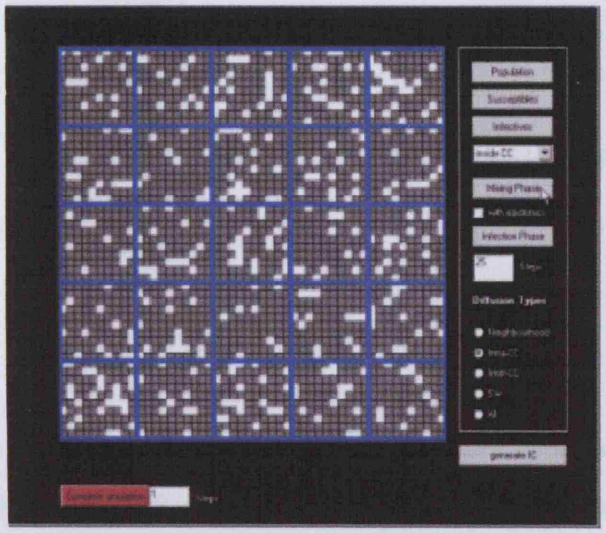


Figure 2-11: CA simulation using a movement rule, that considers only neighborhood movement. This screenshot was taken from the CA implementation of the model, programmed in VB.

The contagion rule is deterministic and totalistic (based on Wolfram's 1022 rule (Wolfram, 2002)). In totalistic rules, the new state of the cell is a function of the states of the neighborhood as we can see in (2.21).

$$(S_{i,j})^{t+1} = f\left\{(S_{i,j})^{t}, (S_{i-1,j})^{t}, (S_{i+1,j})^{t}, (S_{i,j-1})^{t}, (S_{i,j+1})^{t}, (S_{i-1,j-1})^{t}, (S_{i+1,j+1})^{t}\right\} \tag{2.21}$$

The S states of a cell can be expressed as $S = 0, 1, 2$ (susceptible, infected, removed). The transition from susceptible to infected is given by 2.22.

$$If\,(S_{i,j})^t = 0 \Rightarrow (S_{i,j})^{t+1} \begin{cases} 1 & \text{if } \sum_{l=i+1}^{l=i-1} \sum_{c=j+1}^{c=j-1} s_{l,c} \geq 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.22}$$

The transition from infected to removed is given by (2.23) as:

$$If\,(S_{i,j})^t = 1 \Rightarrow (S_{i,j})^{t+1} = 2 \tag{2.23}$$

There is no state transition for a removed individual as (2.24) shows.

$$If\,(S_{i,j})^t = 2 \Rightarrow (S_{i,j})^{t+1} = 2 \tag{2.24}$$

Rule (2.22) states that if at least one of the neighbors of a susceptible is infected, the individual gets infected. Rule (2.23) states that the transition to a removed state depends only on the individual state in the previous time step. Rule (2.24) states that a removed individual stays removed throughout the simulations, which means there is no possible re-infection.

This model was implemented in VB using ESRI Map Objects[14] for the GIS functionalities. Its user interface and display capabilities are indicative of the behavior of the model under different initial conditions (IC). Figure 2-12 presents some time steps of a simulation which considers an initial group of infected individuals inside a σ unit. This may represent a real situation where a group of people were infected, inside a city. The σ unit that contains the infected individuals was chosen randomly, and the

[14] http://www.esri.com/software/mapobjects/index.html

distribution of the individuals inside it also follows a random distribution. The yellow cells stand for susceptible individuals, the red for infectious and the black for removed. Once an individual is removed it can not become susceptible again and so the disease is considered extinct when there are no more infected individuals on the grid. The initial distribution of the population is totally random in these simulations.

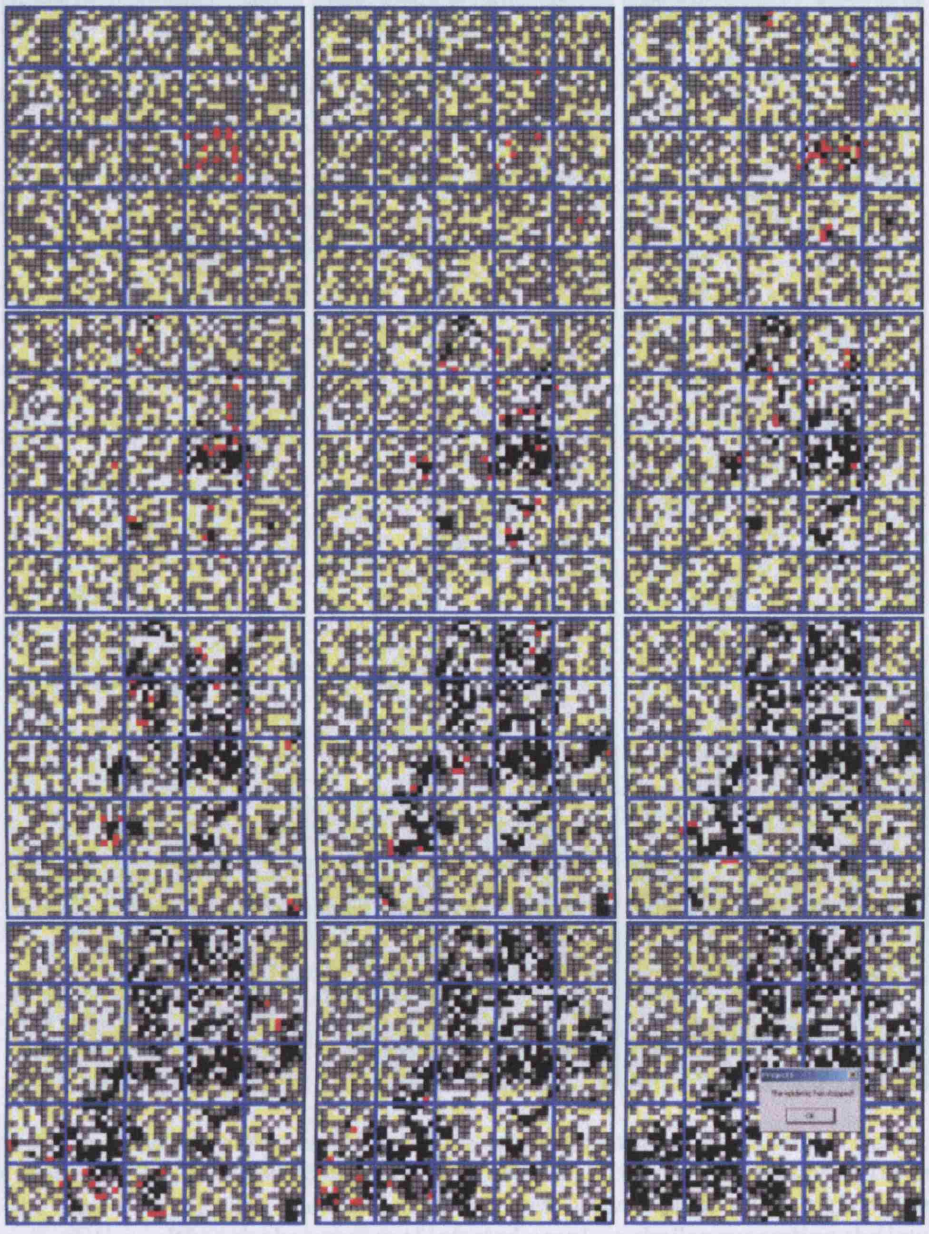


Figure 2-12: Different steps of a simulation, considering as IC a group of infected individuals inside a σ unit (t=0 (top left figure),5,10,15,20,25,30,35,40,45,50,55(bottom right figure)). Since it is a homogeneous space, we can see how infected clusters grow and expand around the infected group.

Figure 2-13 shows another simulation, this time considering an initial random distribution of infected individuals inside the grid and in figure 2-14, the IC is a single infected individual, located randomly inside the grid.

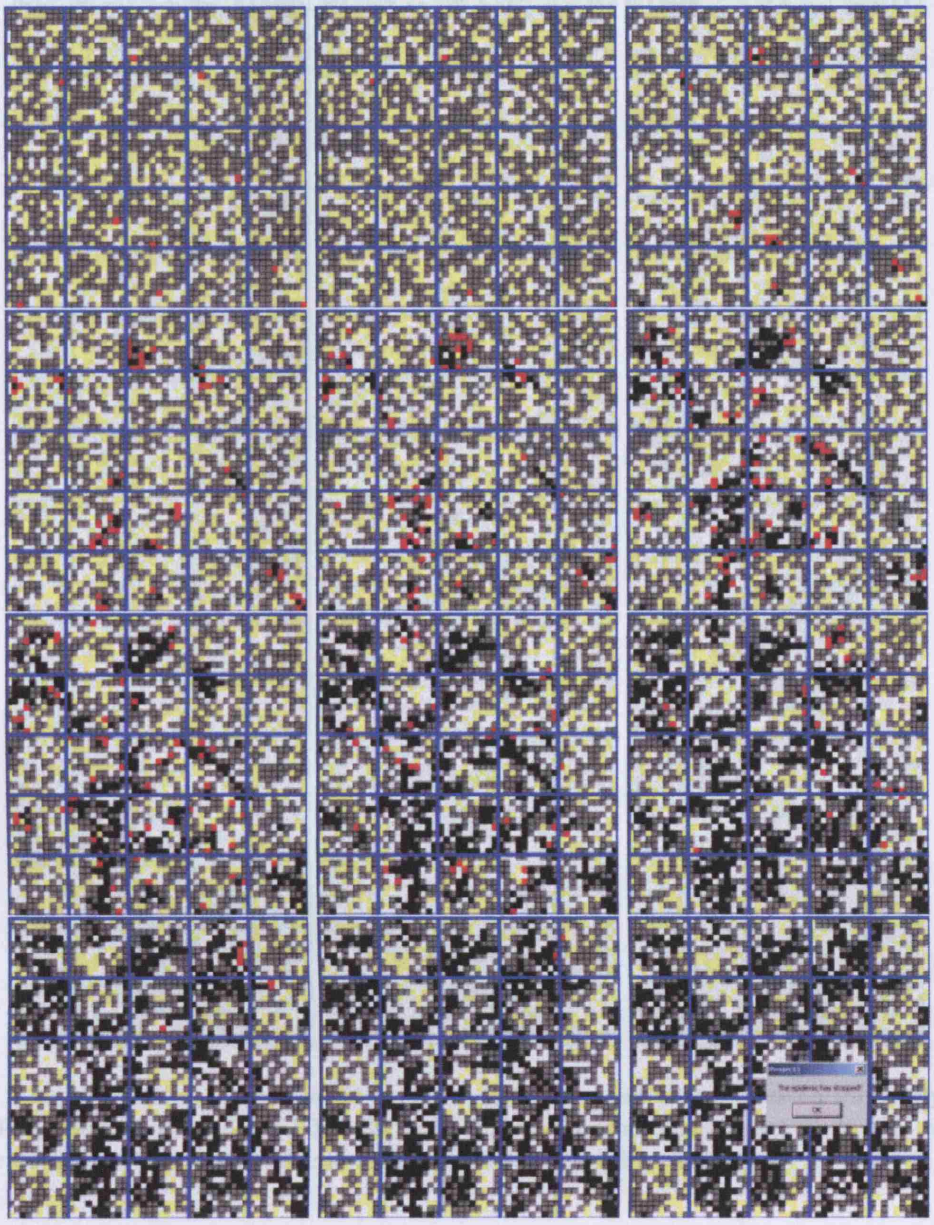


Figure 2-13: Simulation where the IC are a random distribution of individuals inside Λ (t=0 (top left figure),5,10,15,20,25,30,35,40,45,50,55(bottom right figure)). In this case, the outbreak reaches the entire domain and the number of removed individuals is much higher than in the previous simulation, although the number of infected people is similar suggesting the importance of the spatial arrangement.

These simulations give an idea of the dynamics of the model. In the experiments we can see how the diffusion pattern of infection changes according to the initial arrange-

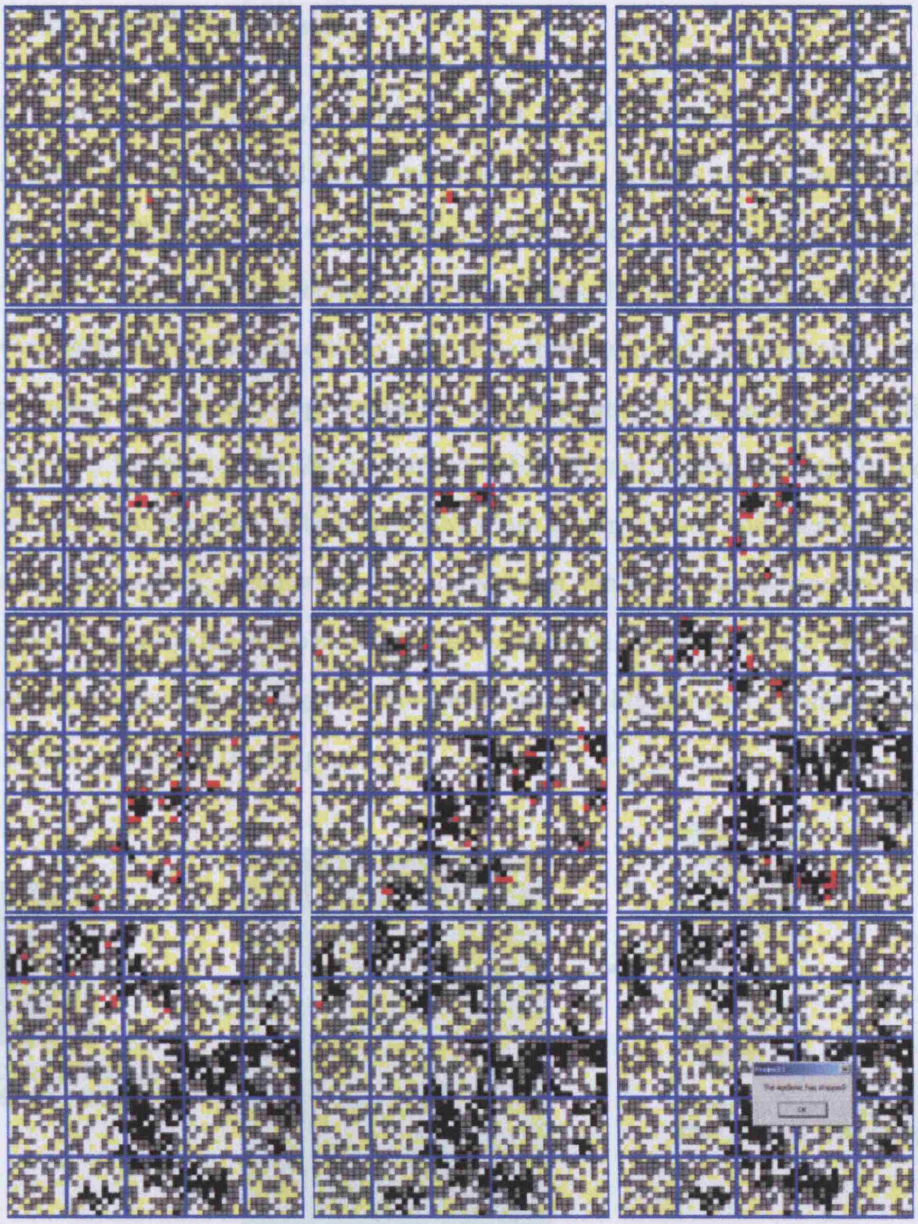


Figure 2-14: Simulation where the IC are a single infected individual, located randomly inside Λ (t=0 (top left figure),5,10,15,20,25,30,35,40,45,50,55(bottom right figure)). In this case, we can see more clearly how the infection spreads through growing clusters.

ment of infected individuals. It should also be note the different patterns that occur for removed individuals at the end of each of the three simulations. These kinds of scenarios which are spatially complex, are easy to create in this kind of simulation model, but would be very difficult to build this simulation using in a deterministic approach.

2.3.2 The ABM Approach

One big disadvantage of the CA model is that it is tied to a fixed and rather inflexible representation of space: grid-like, or as it is defined in GIS terms, as a raster (see section 3.4, page 92). In the specific phenomena we study, this is a poor representation of space, for it assumes homogeneity in equal neighborhoods, something that does not reflect very well real world conditions. On the other hand, considering that one of the initial purposes is a realistic representation of movement, this is also made difficult by the structure of the model, for the movement in SECA is processed by steps in a discrete space. Unlike what happens with CA, ABM space is continuous and location is explicit as we see in figure 2-15; this means that individuals can be simulated independently of the environment, allowing us to specify rules focused on the individuals, rather than on the space itself. For simulating the movement of individuals, this methodology seems to be a better approach to the problem.

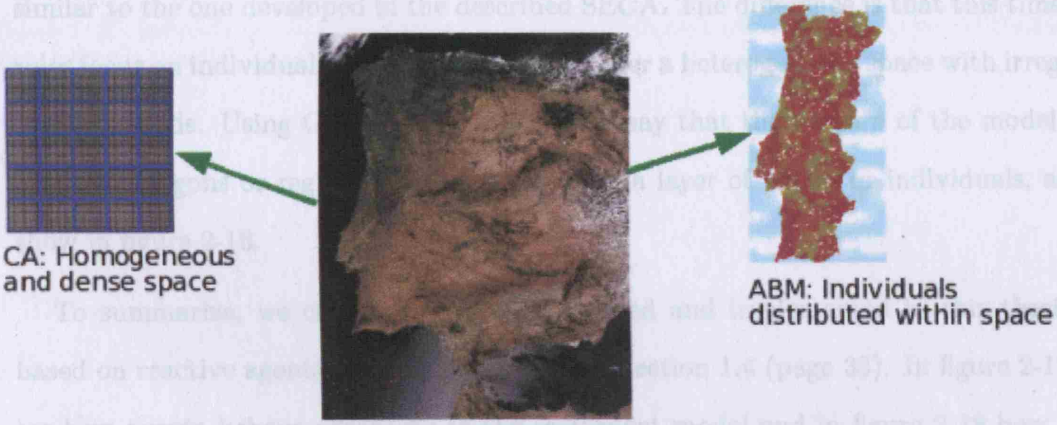


Figure 2-15: The diagram shows the distinct CA and ABM representations of the world. CA focus on space, while ABM focus on individuals.

Although ABM can be based on cellular space (Batty and Jiang, 1999; Reynolds et al., 2001), we can link historically CA and ABM to GIS raster and vector data models. At this light, the disadvantages of ABMs also reflect some disadvantages of the vector format. According to Conolly and Lake (2006) these are:

- It is computationally demanding in terms of storage since each vertex and node

has to be stored in computer memory.

- The manipulation of data is computationally intensive and this also has a cost in terms of Computing-Processing Unit (CPU) time.
- It deals badly with fuzziness and it can lead to artificial precision, although in some cases this can be overcome by using a special structure, a Triangular Irregular Network (TIN).

The first two disadvantages lead to more programming and to the development of optimization techniques. However this is compensated by the realistic representation of reality which is visible in the quality of the mapped outputs. Conolly and Lake (2006) also note the fact that vector data can easily be linked to an attribute data record which makes them very popular in GIS applications that seek database integration. Considering these arguments, the epidemic model was thus implemented as an ABM. The model still follows two phases (mixing and infection) and the movement model is similar to the one developed in the described SECA. The difference is that this time the rules focus on individuals moving continuously over a heterogeneous space with irregular neighborhoods. Using GIS terminology, we can say that the domain of the model is a layer of polygons or regions, and the agents are a layer of points or individuals, as we show in figure 2-16.

To summarize, we can say the model adopted and implemented in this thesis, is based on reactive agents, as we defined them in section 1.4 (page 33). In figure 2-17 we see how agents behave according to the movement model and in figure 2-18 how they behave according to the infection model. These two models are run sequentially at each time step and the infection model performs a simultaneous update through the entire population.

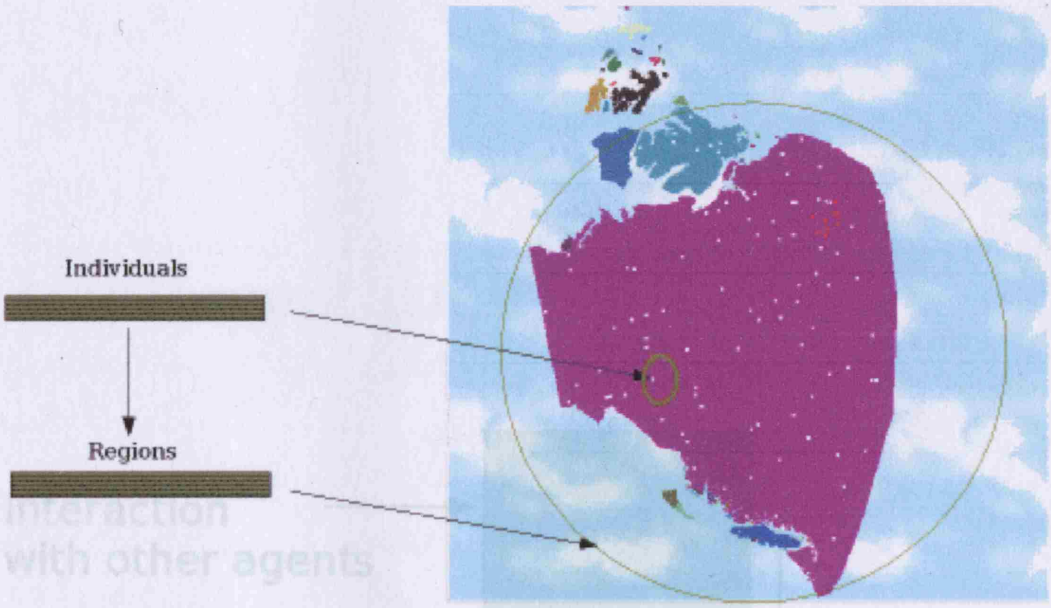


Figure 2-16: Individuals (in a point layer) interacting with each other and with the environment (in a polygon layer), according to rules.

Figure 2-17: The movement model for the displacement of individuals. At each time step, a new position is generated, according to model parameters and the individual is relocated according to it.

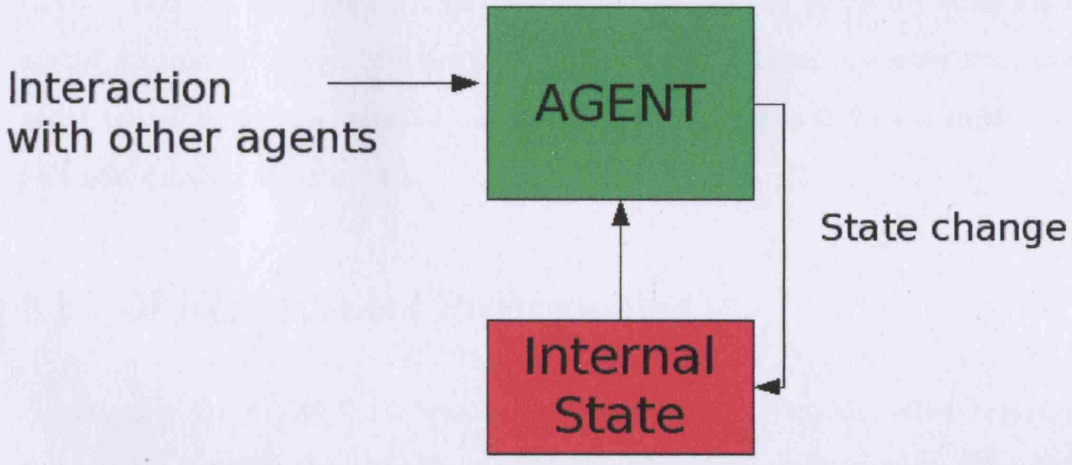


Figure 2-18: The infection model acting on the internal state of the individual, based on outside events (interaction with other individuals) but also on its internal state.

Chapter 3

Implementation

3.1 EpiSIM

Since the only way to demonstrate the theories introduced in section 2 (page 49) is through building a computer model, a considerable amount of time in this project was spent on implementing the model as a computer program called EpiSIM (Epidemic Simulation). This chapter starts with an introduction to programming, giving emphasis to the Object Oriented (OO) approach which was the paradigm adopted in this study. There will be a review of some key concepts that enable us to understand the next section where EpiSIM is explained through its data structures and algorithms. The GIS functionalities - which are an important issue in a spatial model - and the Input/Output (I/O) details are also described as these algorithms can be useful for other kinds of spatial models. Although it is not part of the implementation, the collection, storing and retrieval of data is essential for the program, and thus follows a section about technical details related to this.

3.2 Object Oriented Programming

According to Booch (1993), software presents an inherent complexity which is related to the problem domain, the difficulty of managing the development process, its potential flexibility, and problems of characterizing the behavior of discrete systems. There are five attributes which can be used to describe a complex system:

- The choice of what components are primitive which is relatively arbitrary.
- Intra-component linkages which are stronger than inter-component linkages.
- Hierarchic systems which are composed of diverse subsystems, with different arrangements.
- A complex system that works, evolved from a simple system that works, which implicates starting the design of a complex system from a simple one.

These attributes which Booch (1993) used to describe software can be easily transposed into the context of agent-based modelling and the context of a complex system which

is the target of this study, based on human populations interacting in geographic space (see section 1.4, page 33). The canonical form of such a complex system represents the common abstractions and mechanisms that greatly facilitate the understanding of these systems. Virtually all complex systems take the same canonical form: two orthogonal hierarchies based on class structure and object structure. Considering this, successful software systems will be the ones with a good object and class structure which embody the five attributes of complex systems referred to before. Unfortunately the human brain has some limitations in dealing with complexity and thus there is need for methods to "bring some order into this chaos". These methods include decomposition, abstraction and hierarchy, and it is in this context that the object oriented approach arises. According to Booch (1993), most design methods in software engineering can be categorized as:

- Top-down.
- Data-driven.
- Object-oriented.

Top-down design is based on algorithmic decomposition, also called *waterfall* model, which requires the completion of the analysis phase before the design, and the completion of the design phase before the construction. As the design is based on processes that often change, for instance as a result of new requirements or error corrections, this method has the risk of carrying forward an unstable design. Also, because process decomposition is oriented to solving a specific problem, the resulting design is unique, making it difficult to reuse (Quillin, 2001). All these facts result in top-down design being inappropriate for dealing with complexity. Object oriented design approaches software systems as collections of cooperating objects and treats each object as an instance of a hierarchy of classes. This is the best suited approach to deal with complexity (Booch, 1993). Although these approaches are not directly linked to languages and it is possible to have a top-down design using an object oriented language, it makes more sense to use a language whose topology reflects the design method chosen. In this perspective, it is reasonable to use an object oriented language (like C++) for an

object oriented approach although this is not mandatory. In the following paragraph, we will describe briefly the evolution of programming languages, which also reflects the evolution of ways of thinking and addressing problems.

The first approaches to programming were based on machine code and assembler. However, building a program with these languages has serious disadvantages, for the code is different for every computer and the instructions are very detailed, making programming a difficult task (Skansholm, 1997). The high level languages that arrived on the 1950's/60's facilitated the development of code, where FORTRAN (1954) and COBOL (1959) are examples of this; however, these languages were still rudimentary with the first one presenting a poor structure and poor I/O and the second one, which is finance/administrative oriented, leading to voluminous code. In 1960 a language called ALGOL appeared which evolved into two interesting languages: SIMULA (1967), the first object oriented language, and PASCAL (1971), a non modular language with a good programming structure. In the early 1970's at AT & Bell Laboratories, C was developed as a high level language that enables low level programming (Skansholm, 1997). C provides the freedom to control the computer in detail, allowing descent into the hardware level. About this time, the software industry faced a crisis related to the structured programming paradigm for this approach forces a set of rules on how a good program should be written. It was discovered on the one hand, that well structured programs are not enough for mastering complexity, and on the other hand that the modular approach, with objects, is more successful in projects that involve teams of several people. SMALLTALK (1980) and ADA (1983) were two object oriented languages that arrived in this context, but by far the most popular one is C++ (1985), that derives from C, and was also developed at the AT & Bell Laboratories.

C++ is a strongly typed language, which means each type of data (such as integer, character, and so on) is predefined as part of the language and all constants or variables defined for a given program must be described with one of the data types. It can be seen as a superset of C that adds support for classes, type checking, overload of functions and constant types among other things (Montgomery, 1994). C++ inherits the advantages

of C in enabling low level programming and directly manipulation of the computer memory. Simultaneously, it has high-level language features which makes it easier to read and write than assembly language. On the other hand, the fact that it is not as easy to understand and that here are not as many automatic checks as in other high-level languages, can be seen as drawbacks (Savitch, 1999). In the literature, many examples can be found of ABMs implemented using java (Crooks, 2006; Callaghan, 2005). Java is more portable than C++ (although not 100 % portable as different platforms fire events at different times and in different order) but that is not an essential matter to simulation, as a model will run most of the times in a single computer. However performance does matter, as we are dealing with large datasets and long time series which classify simulations as computationally intensive tasks. The differences in performance between java and C++ have been widely discussed and some benchmarks, like the one presented by J.P.Lewis and Neumann (2003), show proximity between the two of them in certain tasks such as numerical calculations. Nevertheless there are some reasons why java is, in principle, slower than C++ which are mainly related to its increased memory use. Namely[15]:

- Programs that utilize automatic garbage collection typically use about 50 % more memory than programs that do manual memory management.
- Many of the objects that would be allocated on stack in C++ will be allocated on the heap in java.
- Java objects will be larger, due to all objects having a virtual table plus support for synchronization primitives.

Besides this, although being based in C++, java does not include many features of this language, the most important being the ability to use pointers, and many others such as operator overloading, automatic type conversions, support for structures and multiple inheritance. The lack of this functionalities turn it into a less powerful language that justifies the choice of C++ in the context of this study. In the next paragraphs, there will be a more detailed description of some key object oriented concepts, that will help

[15] http://www.jelovic.com/articles/why_java_is_slow.htm

us to explain the program better. Objects can be seen as a collection of data and instructions for manipulating this data, as we see in figure 3-1. They encapsulate the data structures together with the program code to manage them (Montgomery, 1994).

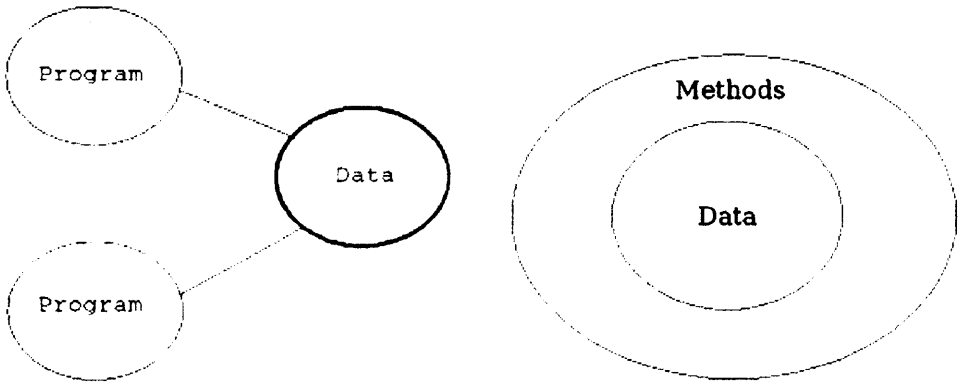


Figure 3-1: The information processing in the traditional model (left) and object model (right). In the traditional model, data is stored outside the program while in the object model, it is encapsulated within the object, together with the methods to manipulate it.

The key difference between the object oriented and the structured approach is that objects are entities and functions rather than procedures, and in this way they tend to closely reflect natural systems in the real world as we see in figure 3-2.

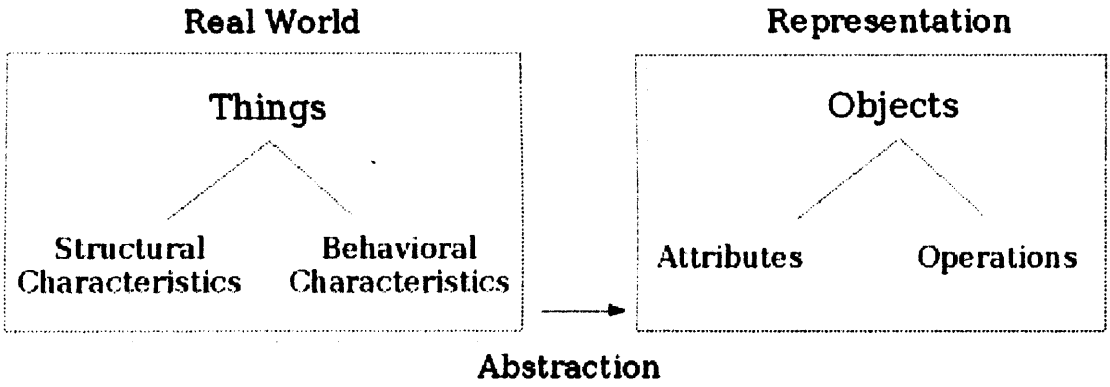


Figure 3-2: Objects as representational constructs of entities (based on Alhir (1998)).

The object oriented programming model implements complex systems through various levels of abstraction. One advantage of using objects is the protection to changing properties via encapsulation and information hiding (Montgomery, 1994). Another ad-

vantage is its re-usability: the procedures in an object can be used in another object through a system of classes, hierarchies and inheritance. Classes, which are represented in figure 3-3, can be defined as descriptions of objects, or a set of objects with a common implementation (Alhir, 1998).

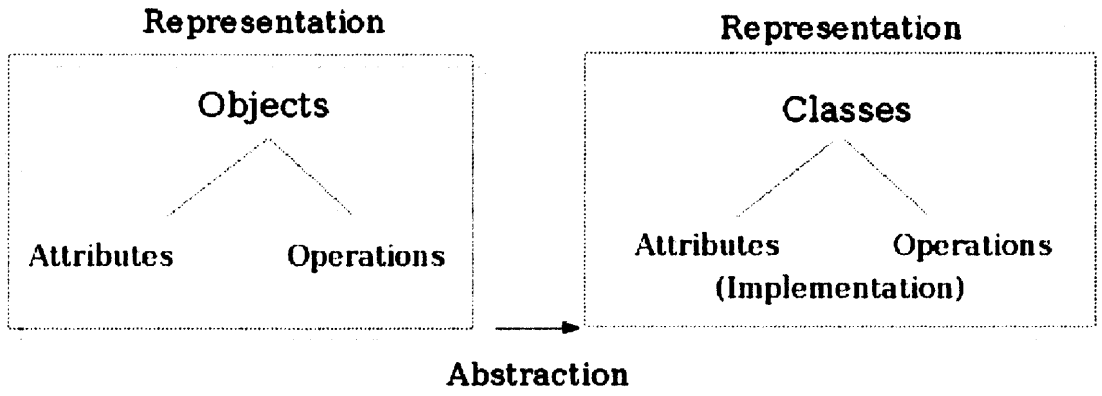


Figure 3-3: Classes as implementations of objects (adapted from Alhir (1998)).

The hierarchic structure of the object oriented model establishes relationships amongst classes, such as generalization or aggregation; generalization supports the inheritance of characteristics from another class or classes (single and multiple inheritance) and aggregation defines a system as composed of subsystems. This introduction to object oriented programming is far from being exhaustive, and there are many concepts that were not introduced here such as association, messages, and polymorphism. However, we intend to only introduce the essential elements for understanding EpiSIM and a more complete and developed view of the object oriented model can be found in the references (for instance, in Montgomery (1994)).

3.3 Programming The Model: Data Structures and Algorithms

The Unified Modelling Language (UML) is a language applied to specifying, constructing and documenting systems (Alhir, 1998). Figure 3-4 shows the notation used in this section for describing and explaining EpiSIM.

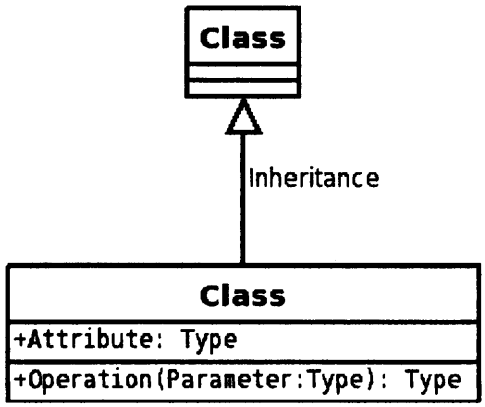


Figure 3-4: The UML notation followed in this study.

Every C++ program starts with a *main*, which is the first function executed when we run the application. Apart from that, in object oriented programming there is no temporal sequence of functions as in structured programming, as the actions are triggered by events. However, it is still possible to present a flow diagram with the files that implement EpiSIM and their relationships as we can see in figure 3-5. The *main* part calls the *canvas* (which is the core of the program) and the *canvas* calls *quadtree*, *shapelib* and *form* for specific functions. There is also a convention in C++ to distinguish between header files with class definitions (*.h*), and implementation files with method definitions (*.cpp*). Following this logic, *canvas*, for instance, is split into *canvas.h* and *canvas.cpp*.

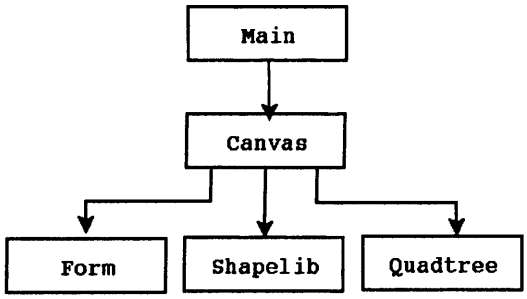


Figure 3-5: EpiSIM implementation through several files, that group different functionalities, and how they relate to each other.

Two sets of libraries were used to build this software: © Qt, for the graphic Appli-

cation Program Interface (API) and Shapelib for accessing geographic data. For now, we are going to concentrate on the description of the first, as the second one will be developed in section 3.4 (page 92), in the context of GIS. Qt is a set of tools, but what is actually used in this program is the classes library, which concerns the user interface (UI). In figure 3-6, we show the *about* box of the program, which displays information about Qt.

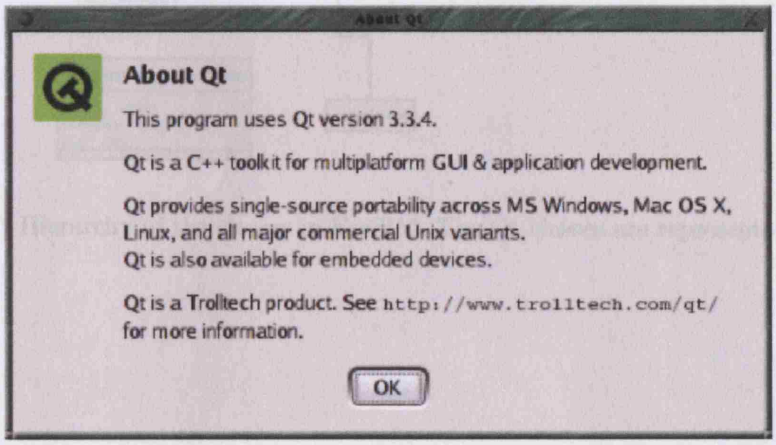


Figure 3-6: Qt *about* box.

Although Qt is open source, it is not free software[16] for there is a charge for commercial use; otherwise, it can be downloaded freely on the web[17]. Figure 3-7 presents the diagram showing the classes used for developing EpiSIM. The Qt classes are represented in green and all the others, created by means of generalization, inheriting from the Qt classes, were implemented for this program. In the first approach, it is possible to identify three major groups of classes: UI, GIS and the ABM.

The user interface (figure 3-8) includes three forms: *FormGrafico1*, *FormGrafico2* and *FormModel*. Each one is implemented as a class and all of them inherit their characteristics from class *QDialog* (figure 3-8, on the right side). Classes *FormGrafico*, *FormGrafico1* and *FormGrafico2* implement the forms containing graphics and *FormModel* is the main form of EpiSIM with all the buttons and menus. Because the list of variables and methods of this class is so extensive, we had to represent it separately in

[16] www.gnu.org
[17] http://www.trolltech.com/

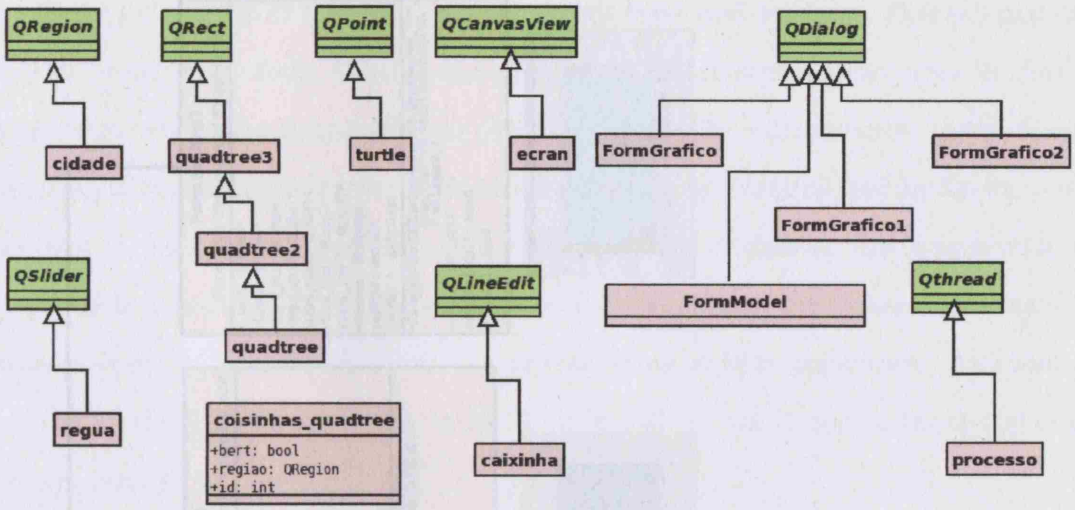


Figure 3-7: Hierarchy of the classes in EpiSIM. The Qt classes are represented in green.

figure 3-9.

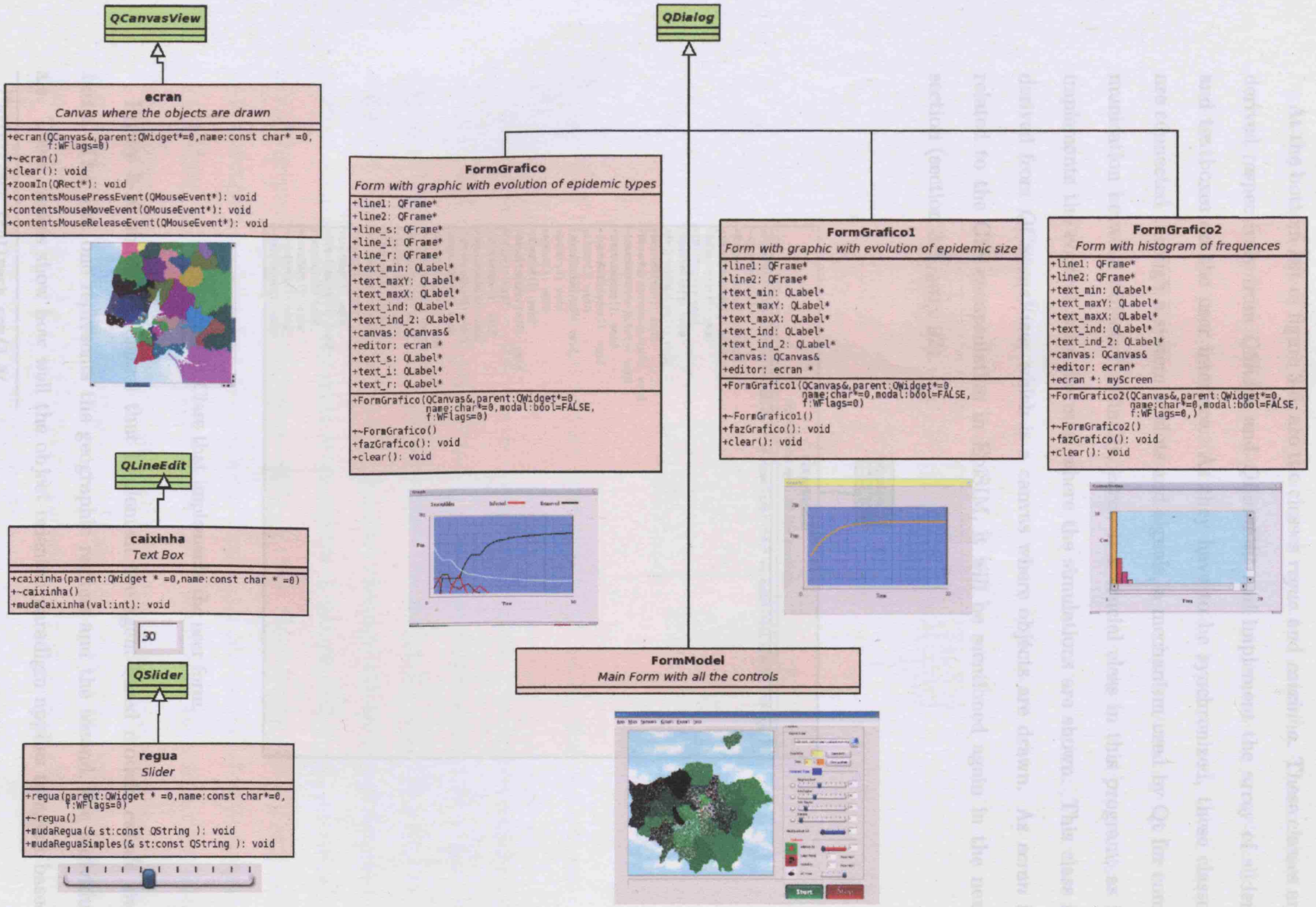


Figure 3-8: Classes that implement the user interface.

At the bottom left of figure 3-8, are the classes *regua* and *caixinha*. These classes are derived respectively from *QSlider* and *QLineEdit* and implement the array of sliders and textboxes on the user interface. As they have to be synchronized, these classes are connected through a system of slots and signals, a mechanism used by Qt for communication between objects[18]. Finally, *ecran* is a crucial class in this program, as it implements the control with the map where the simulations are shown. This class is derived from *QCanvasView*, which is a canvas where objects are drawn. As *ecran* is related to the GIS encapsulation in EpiSIM, it will be mentioned again in the next section (section 3.4, page 92).

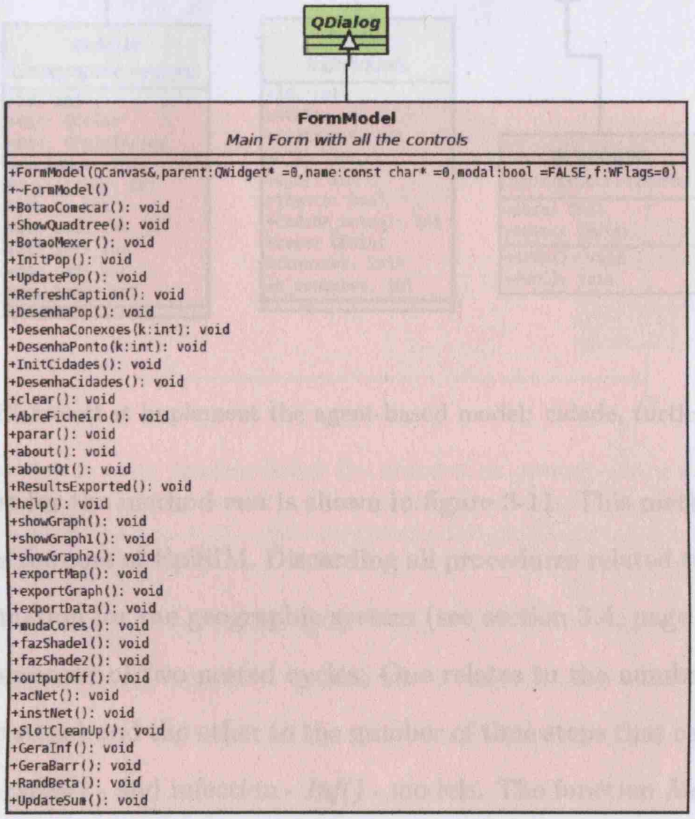


Figure 3-9: Class that implements the user form.

Figure 3-10 shows the classes that implement the agent-based model: *cidade* and *turtle*. The first one represents the geographic regions and the second, the individuals. These classes show how well the object oriented paradigm applies to agent based

[18] http://doc.trolltech.com/3.3/

modelling, a framework which is totally based on the behavior of singular entities with a common structure. The other class present in figure 3-10, *processo*, represents the modelling process itself, and it inherits its characteristics from *Qthread*. *Threads* are a way of splitting EpiSIM into simultaneously running tasks[19]. If the model occupies the main thread, while it is running it is not possible to perform other tasks, like for instance interacting with the user interface; for this reason, the modelling process was created as a separate thread.

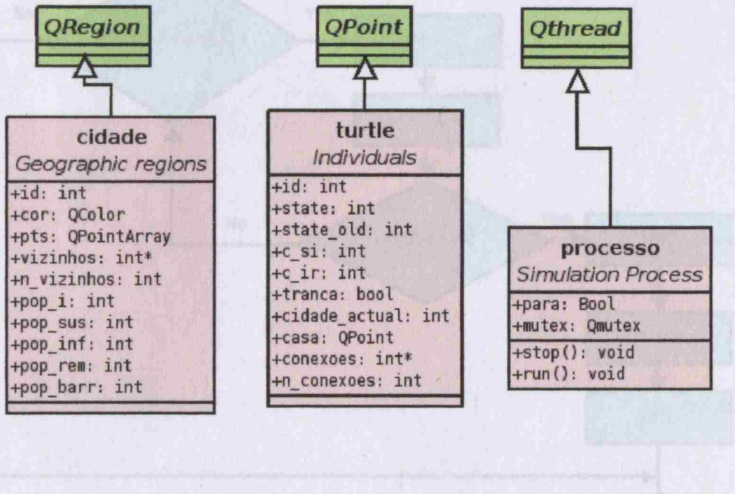


Figure 3-10: Classes that implement the agent-based model: cidade, turtle and processo.

The flowchart for the method *run* is shown in figure 3-11. This method contains the process which is the core of EpiSIM. Discarding all procedures related to the UI and the storing of information for the geographic system (see section 3.4, page 92), each run of the model is composed of two nested cycles. One relates to the number of simulations (as a set of time steps) and the other to the number of time steps that call, sequentially - the movement - *Mov()* - and infection - *Inf()* - models. The function *Mov()* is explained in the flowchart in figure 3-12. Basically, this function reads the probabilities assigned on the *sliders* and runs a cycle through all individuals to animate them with movement. The way the movement is assigned to each individual is generating a random variable and comparing this value with the probabilities of each component of movement, through a chain of nested conditions. This system is perhaps easier to understand by looking at

[19] http://en.wikipedia.org/wiki/Multithreading

figure 3-13 where A,B,C and D correspond to the different components of movement. Once one component of movement is selected, it calls the corresponding algorithm of movement to displace the individual.

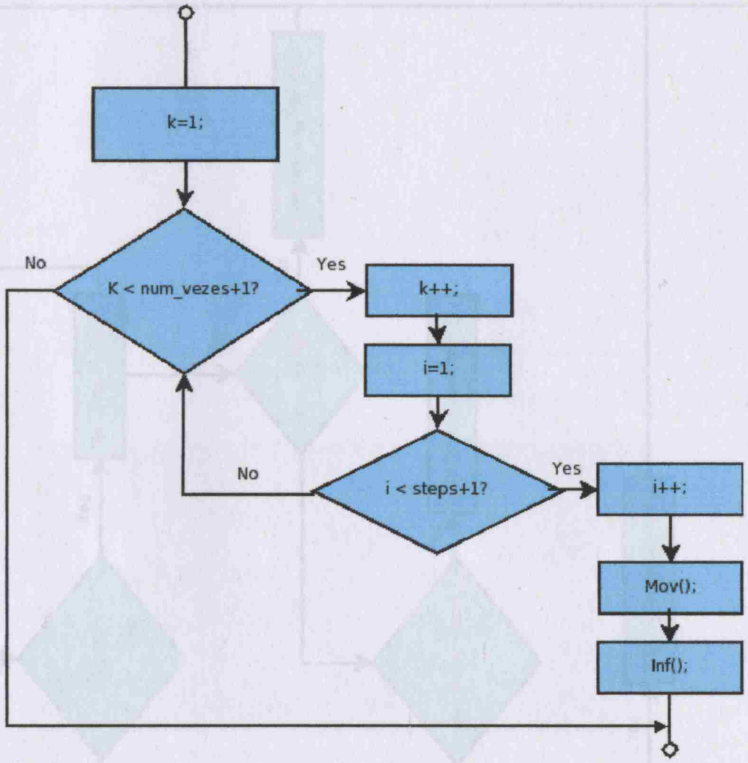


Figure 3-11: The method *run* implementing the simulation process. In this flowchart we see how two nested loops, one for the number of simulations ($k < num_vezes + 1?$) and other for the number of time steps ($i < steps + 1?$), call sequentially the movement and infection models.

The infection model represented in figure 3-14, is a bit more complex than the movement model; to understand this, it is important to look at some of the members of the class *turtle* in figure 3-10. All individuals are members of the class *turtle*; its state is controlled by variable *state*, which can assume the values of: 0 (susceptible), 1 (latent), 2 (infectious) and 3 (removed). Variables *state_old* and *tranca* are in the program to ensure the synchronous update of all individuals. The loop goes through the entire population and targets the latent and infectious individuals. The latent individuals can change state if the latent period is over; the infectious individuals can infect any individual contained in a *buffer* around them (see section 3.4, page 92) that matches certain conditions, and can change state if the infectious period is over.

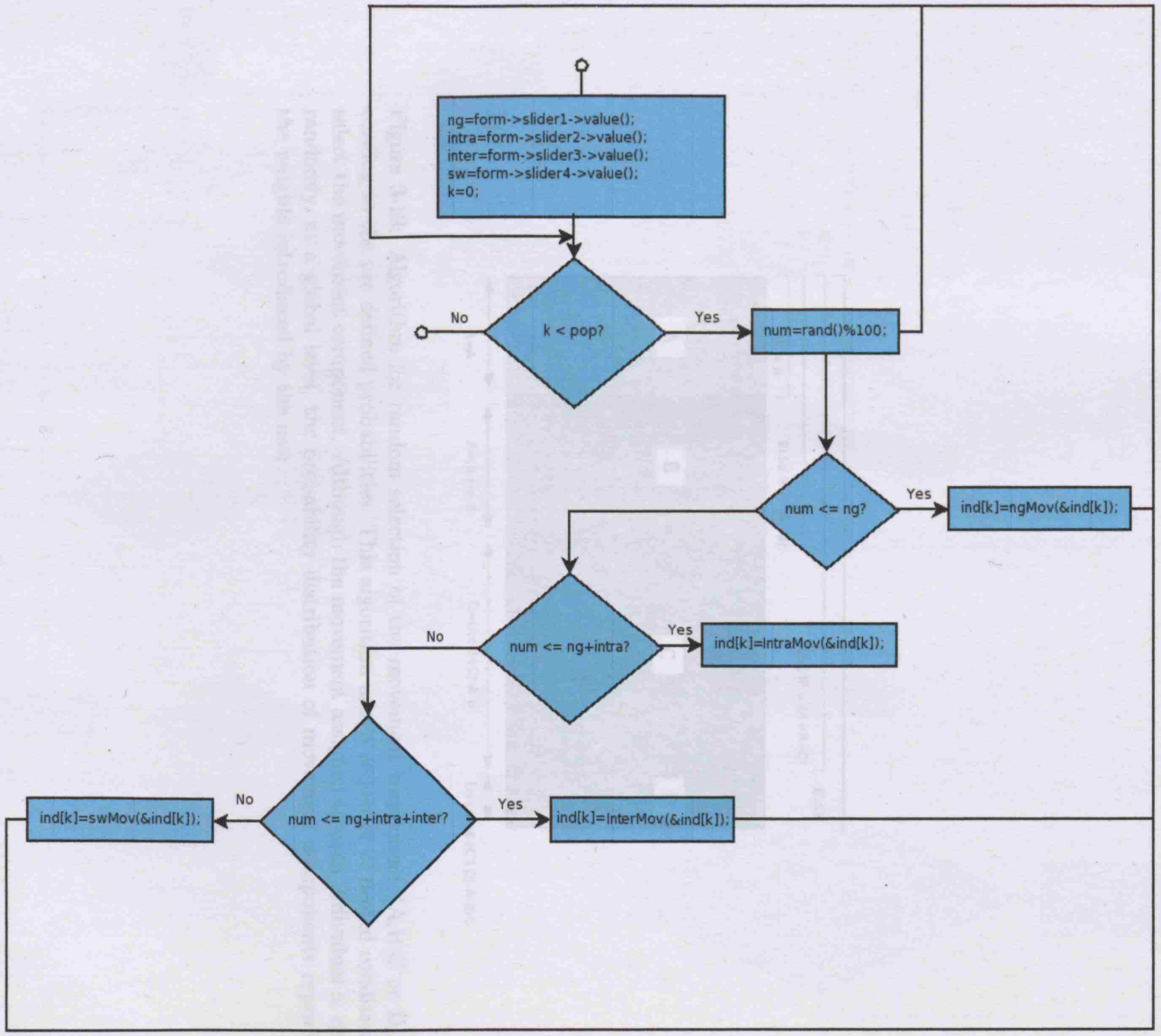


Figure 3-12: Flowchart of the function *Mov()* which implements the movement model. This function, which is located within a loop through the entire population, allocates each to individual to a component of the movement, according to the probabilities introduced by the user. Once the movement component is chosen, the function calls an appropriate function to move the individual.

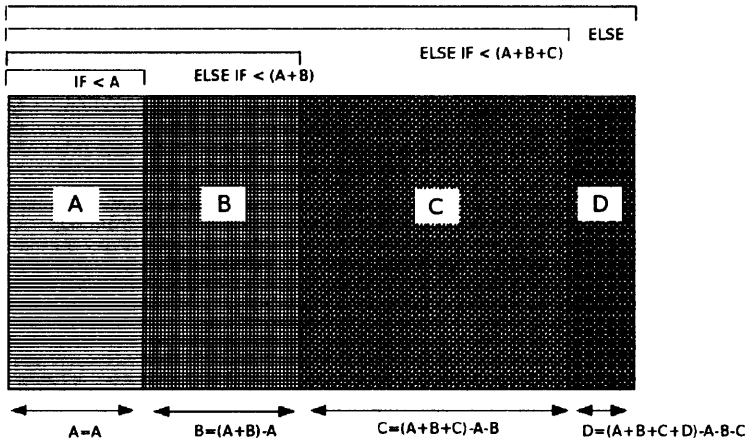


Figure 3-13: Algorithm for random selection of the movement component (A,B,C or D), according to the pre defined probabilities. This algorithm uses a sequence of nested conditions to select the movement component. Although the movement assigned to each individual is chosen randomly, at a global level, the probability distribution of movement components reproduces the weights introduced by the user.

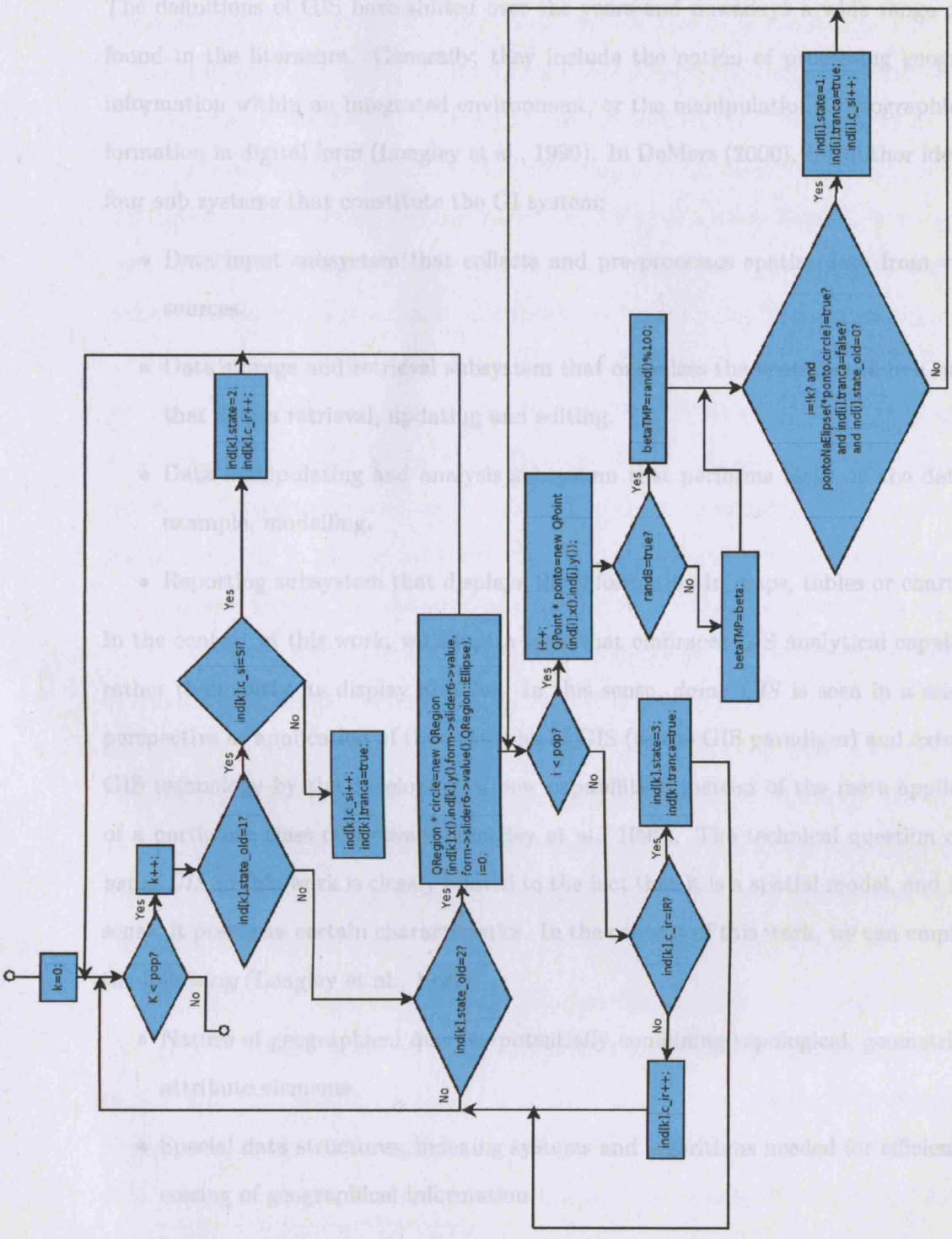


Figure 3-14: The flowchart of the function *Inf()*. This function consists of a loop that goes through the entire population and updates the state of individuals, according to its current state and to counters, that keep track of the length of the latent and morbid periods. When an individual becomes infectious, there is a new loop through the population to check if there is any individual available for infection around it.

3.4 Geographic Information Systems Functionality

The definitions of GIS have shifted over the years and nowadays a wide range can be found in the literature. Generally, they include the notion of processing geographic information within an integrated environment, or the manipulation of geographical information in digital form (Longley et al., 1999). In DeMers (2000), the author identifies four sub systems that constitute the GI system:

- Data input subsystem that collects and pre-processes spatial data from various sources.
- Data storage and retrieval subsystem that organizes the spatial data in a manner that allows retrieval, updating and editing.
- Data manipulating and analysis subsystem that performs tasks on the data, for example, modelling.
- Reporting subsystem that displays the information in maps, tables or charts.

In the context of this work, we adopt a view that embraces GIS analytical capabilities rather than solely its display abilities. In this sense, *doing GIS* is seen in a scientific perspective as application of the principles of GIS (or the GIS paradigm) and extending GIS technology by the developing of new capabilities, instead of the mere application of a particular class of software (Longley et al., 1999). The technical question of *why using GIS* in this work is closely related to the fact that it is a spatial model, and in this sense, it possesses certain characteristics. In the context of this work, we can emphasize the following (Longley et al., 1999):

- Nature of geographical queries, potentially combining topological, geometric and attribute elements.
- Special data structures, indexing systems and algorithms needed for efficient processing of geographical information.
- Multi-dimensional nature of geographical information.
- Voluminous nature of much geographic information.

The option of building a GIS application from scratch, rather than using an existing desktop GIS is related to issues of efficiency and flexibility. There is still a lack of integration of GIS with other applications, namely numerical simulation models, and the customization of a desktop GIS with high level user oriented languages is slower than core programs created with a low level language such as C++ (Longley et al., 1999). In fact, the current trend is that GIS is becoming application niche software, and some application niche software includes GIS (Longley et al., 1999); that is, if a map or a piece of GIS functionality makes sense within an application, it will then include it, rather than refer to a proprietary software GIS. At this point, it could be argued that the integration within an open-source GIS such as GRASS or Quantum GIS[20] would provide a good framework for this model with the advantage of being fast, as it be would compiled as C++ native code. However, the full control of the algorithms and therefore of the performance, as for instance in the programming of the quadtree for spatial indexing, in section 3.4 (page 92), can only be achieved when programming a software from scratch. We can say that the final application developed in this study fills the requisites of a true desktop GIS. Figure 3-15 shows the different levels of operations effectuated in EpiSIM. The diagram on the left is a functional scheme and focuses on the different stages of the GIS: reading, storing and displaying data, with the model fully embedded in the system and using its tools to operate over the data. The diagram on the right is a more technical view of these functionalities and it also suggests pre-processing of data before it enters the program in contrast to the post-processing of the output, outside the program.

Today there is a wide range of plugin software that can help in building a stand-alone GIS using a generic language. Most commercial solutions rely on the COM or .NET technologies from Microsoft[21] and provide OCX objects that encapsulate GIS functionalities, generally for the analysis and display of geographic information. There are, as well, some open source alternatives, namely the *Terralib* classes[22]. Unfortunately when

[20] http://qgis.org/
[21] http://www.microsoft.com/com/default.mspx
[22] http://www.terralib.org/

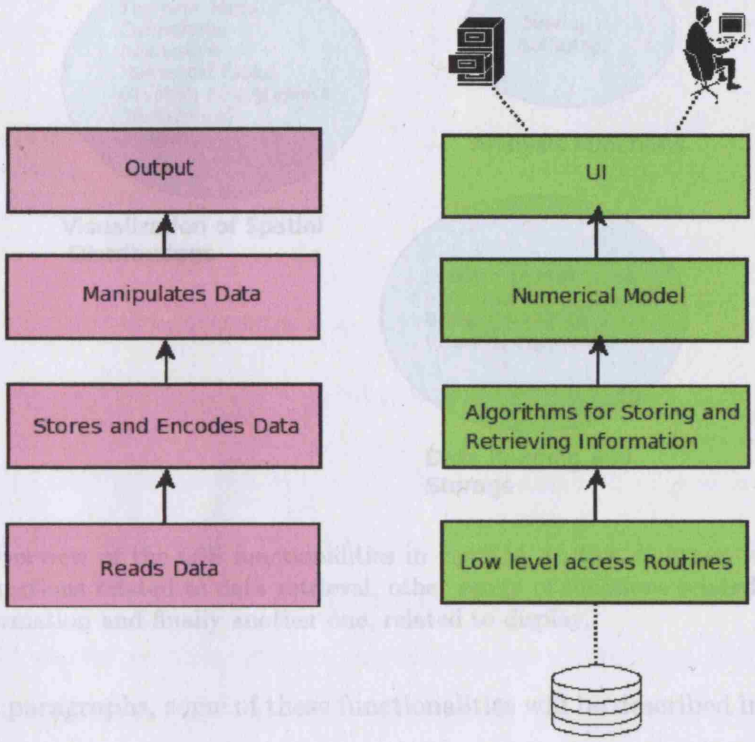


Figure 3-15: Structure of the GIS developed in EpiSIM. On the left we see a diagram with the sequence of operations with data and on the right, there is a more detailed view which includes the low level routines and the user interface. The GIS is embedded in all these parts of the program.

this program was first developed, these classes did not offer any display capabilities. The approach followed in this study was to develop all these functionalities from scratch, avoiding extra cost and also maintaining full control and knowledge over all the operations. In this way, the system is completely transparent for the programmer, which happens as well with open source libraries, but not with commercial solutions. The only external help concerns the low level access to data, in this case the *Shapefile* coverages. This part of the program, although fully incorporated in the code, was coded based on a set of open source libraries called *Shapelib*[23] . Figure 3-16 shows an overview of the different GIS functionalities developed in EpiSIM. It is important to note that this only includes the GIS aspect of the program and excludes all the model functions that make use of these tools.

[23] `http://shapelib.maptools.org/`

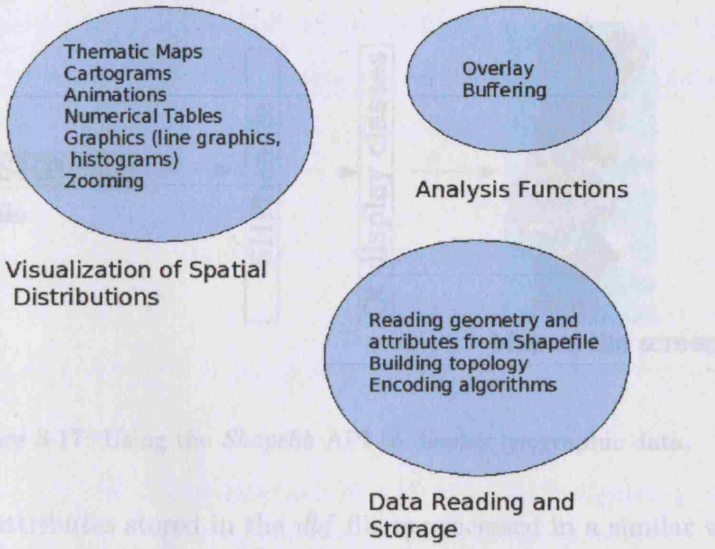


Figure 3-16: Overview of the GIS functionalities in EpiSIM. In this diagram, we can identify one group of functions related to data retrieval, other group of functions related to analysis of geographic information and finally another one, related to display.

In the next paragraphs, some of these functionalities will be described in more detail. The first question faced in the development of this application was the choice of a GIS data format. As discussed earlier in section 2.3 (page 63), we decided to use a vector data model, so the choice went to the widely known *Shapefile*, a working and interchange format promulgated by ESRI for simple vector data with attributes. The reasons behind this choice are that it is a very common format which makes EpiSIM more usable. It is also an open format, and the all the technical information necessary to write a program that uses *Shapefiles* is fully described in ESRI's whitepaper[24]. The Shapefile C Library, also known as *Shapelib*, "provides the ability to write simple C programs for reading, writing and updating (to a limited extent) ESRI Shapefiles, and the associated attribute file (.dbf)"[23]. The way *Shapelib* gives access to *Shapefiles* is reading the data into a structure called *SHPHandle* that provides functions to access all the individual shapes. This data is stored inside EpiSIM and displayed on the screen using the *Qt* graphical classes, as described on 3.3 (page 81). In figure 3-16 we can see a diagram with the components in this process.

[24] `www.esri.com/library/whitepapers/pdfs/shapefile.pdf`

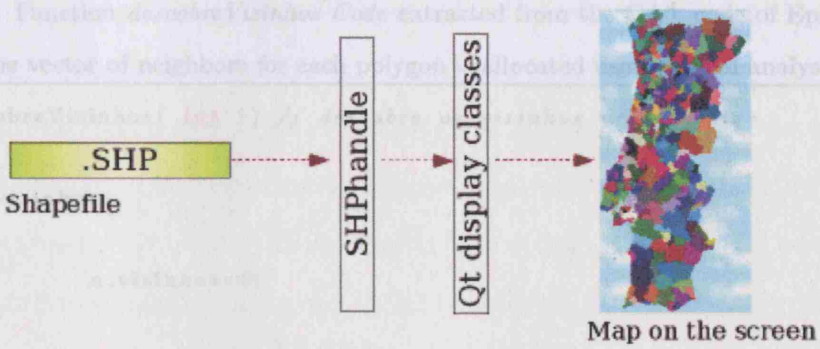


Figure 3-17: Using the *Shapelib* API to display geographic data.

The access to attributes stored in the *dbf* file is processed in a similar way, through a structure called *DBFHandle*; more details about this API can be find in the *Shapelib* website[25]. More than simply displaying geographic information, there was the need for storing and retrieving these data in an efficient form. As *Shapefiles* lack explicit topology (DeMers, 2000), it is important to build some topological information that will save time in some operations required by the model, such as information stored directly into objects: neighborhoods of a polygon (calculated by adjacency) and point-in-polygon operations are examples of this. An example of neighbors of a given polygon is shown on figure 3-18, where these are evaluated by the function *descobreVizinhos* listed as 3.1.

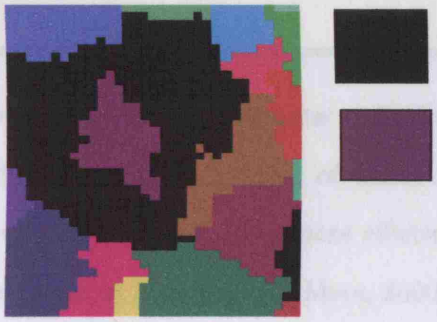


Figure 3-18: An example of the neighbors (in black) of a polygon (in violet) as they are calculated in EpiSIM.

[25] www.esri.com

Listing 3.1: Function *descobreVizinhos Code* extracted from the C++ code of EpiSIM. In this function, the vector of neighbors for each polygon is allocated using spatial analysis operations

```
void descobreVizinhos( int i) // descobre os vizinhos desta regiao
{
int j, n_vizinhos;

                n_vizinhos=0;

                for( j = 0; j < n_poligonos; j++ ){


                    if ( (QRegion(cidades[i].pts).boundingRect().intersects(
                            QRegion(cidades[j].pts).boundingRect()) ) &&
                        (QRegion(cidades[i].pts)!=QRegion(cidades[j].pts)==true)
                        ) {

                            cidades[i].vizinhos[n_vizinhos]=cidades[j].id;
                            n_vizinhos++;
                        }


                }
                cidades[i].vizinhos = (int *) realloc (cidades[i].vizinhos,
                    n_vizinhos*sizeof(int));
                cidades[i].n_vizinhos = n_vizinhos;

}
```

Some differences between raster and vector data were already covered in section 2.3, (page 63): one major difference is the encoding of data. In fact, the algorithms for storing raster data are much simpler and much more efficient than the ones applied to vector data. Vector models such as *Spaghetti* (DeMers, 2000) do not store the topology and topological models possess increased processing overheads. On the other hand, raster data can make use of some very efficient and sophisticated compacting algorithms. We are going to look at some of these algorithms in a little more detail, for they were the inspiration for the vector storage method implemented in this study.

Encoding using *Run-length codes* is an old fashioned method where each cell is given a numerical value corresponding to a categorical data. Analyzing row by row, it is possible to store runs of data (that is, sequences in which the same data value occurs in many consecutive data elements) as a single data value and count: these are called the run-length codes. In *Raster Chain Codes*, rather than operating on a row basis, it is possible to store chain codes in any direction, storing the number of grid cells and the vector direction. Finally, the *Block Codes* and *Quadtrees* rely on a collection of square grid cells as the primary unit of storage (DeMers, 2000). *Block Codes* are similar to *Run-length codes*, but instead of giving start and end points, a square group of cells is selected; we thus can see them as 2D *Run-length code.*

Before describing the other indexing method, it is worth introducing the *Tree* data structure. A *Tree* is a nonempty finite set of labeled nodes beginning at the root node. Each node is either a leaf or an internal node as we show in figure 3-19, on the left. An internal node has one or more child nodes and is called the parent of its child nodes. The *KD-tree* is a multidimensional search *Tree* for points in a k dimensional space[26]. Each internal node of the *KD-tree* contains one point and also corresponds to a rectangular region; the root of the *Tree* corresponds to the entire region. In the 2-D case, the *Tree* is searched downwards for the leaf corresponding to the rectangle containing the point, and then this leaf is split into two in order to accommodate the new point (figure 3-19 on the right). This approach has the disadvantage of forcing the shape of the *Tree* to depend on the order in which points are inserted. The *adaptive KD-tree* solves this problem by choosing a splitting point which divides the set of points into two equal-sized sets (Longley et al., 1999).

There are other kinds of *Trees* such as the *KDB-tree* and the *BSP-tree* (Longley et al., 1999). For this work, the more relevant one is the structure called the *Quadtree.* The term *Quadtree* refers to all kinds of trees that are build by the recursive division of space into four quadrants (Longley et al., 1999). In other words, the domain is successively divided into uniform square groups of grid cells with the same attribute

[26] `http://www.nist.gov`

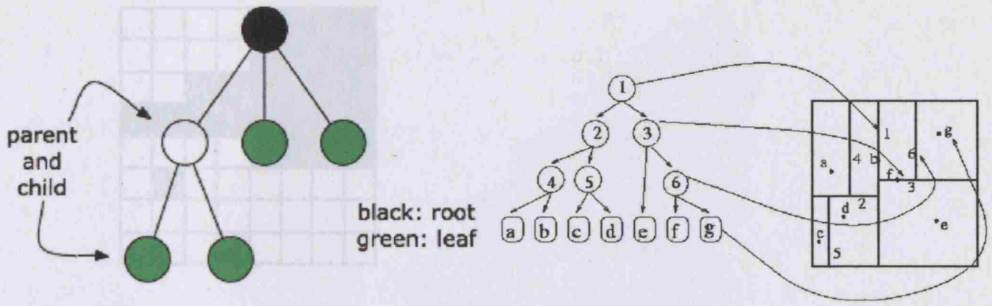


Figure 3-19: *Tree* Data Structure with different types of nodes(left) and *KD-Tree* used to search the corresponding 2D image (right)[27].

[27] http://groups.csail.mit.edu/graphics/classes/6.838/S98/meetings/m13/kd.html

value. The *Point Quadtree* resembles the described *KD-Tree* with the difference being that the space is split into four quadrants, typically following the cardinal directions instead of two. Another variant, the *PR Quadtree*, does not use points of the data to divide space but at each level, it divides the area into four equal sub-squares (Longley et al., 1999). The example in figure 3-20 shows a monochrome image (a bitmap) and the corresponding *Quadtree*. Nodes are labeled b, w, or g, for black, white or grey. If all the pixels covered by a node have the same color (black or white), there is no need to have children under that node. This allows us to compress images, especially when the image contains many large homogeneous regions.

Figure 3-21 shows a practical example of the decomposition of a map using a *Quadtree*.

The *Region Quadtree* is particularly interesting for this work for it stores a rasterised approximation of a polygon and in this way, it is possible to apply the concepts described above to vector data. In this case, the area of interest is enclosed by a square, which is repeatedly divided into four squares of equal size until a square is completely inside the polygon or the maximum depth of the tree is reached (Longley et al., 1999). The *Quadtree* implemented in this study works as a spatial indexing system to search for a particular polygon in a collection of polygons which constitutes the *Shapefile*. As mentioned earlier, this data format does not include topology and thus there is a need to create a system to speed up spatial queries without querying the entire set. This is very important as the agent-based model intensively uses polygon search operations.

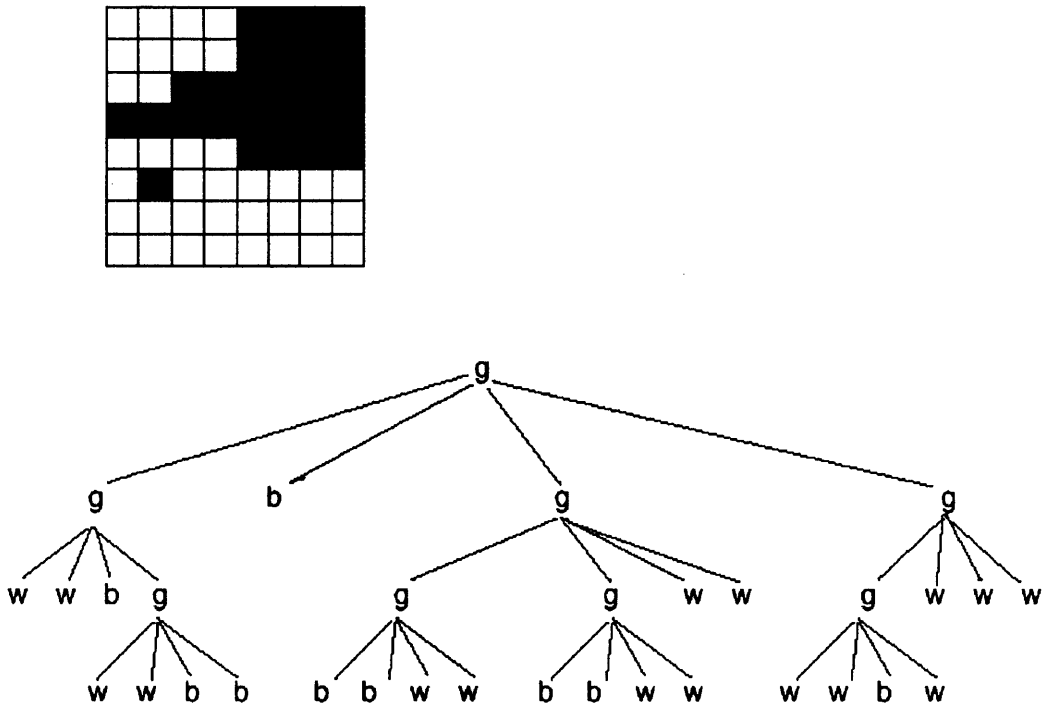


Figure 3-20: A *Quadtree* (bottom) used to search the black and white image (top)[28]. Note that a level of the tree that includes both, black and white pixels is called grey (g).

[28]http://www.dgp.utoronto.ca/~mjmcguff/courses/csc191/02s/assignment2/assignment2.html

This structure was created with a finite number of levels, three, each level corresponding to a rectangular division of space. The last level stores a collection of polygons that will be searched till it matches the criteria as in figure 3-22.

The *Quadtree* was implemented as a recursive class shown in figure 3-23. This object is instantiated when the *Shapefile* is open through the function *criaQuadtree(...)* and it is queried through *estaNaQuadtree(...)*.

There is one problem with this algorithm which is the fact that the polygons might not be fully enclosed by the rectangles, in which case they must belong to two different collections, thus increasing the storage space. This would obviously never occur in *raster* space. Still this technique proved to be very effective: on a 100 polygon layer with 9 polygons in the last level of the *Quadtree*, the slowest search takes 17 steps (4+4+9) against 100 on a non optimized search which is about 5 times faster.

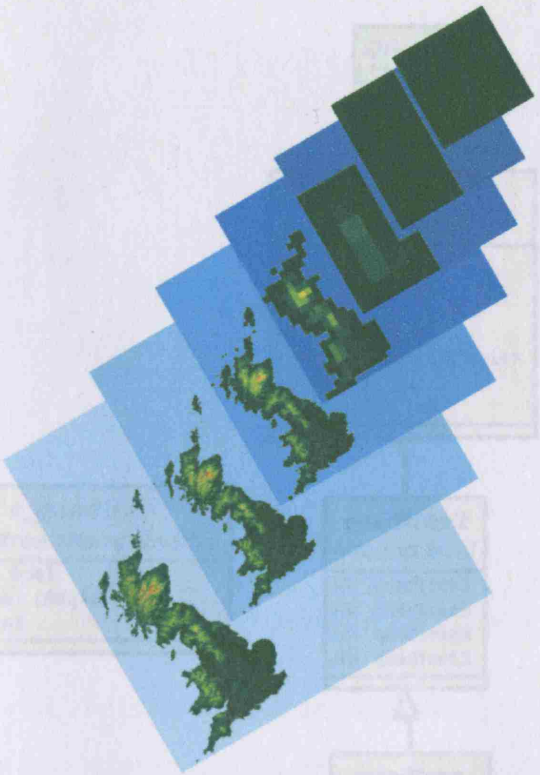


Figure 3-21: Illustration of a *Quadtree* used to decompose a map[29]. From bottom to top, we can see the different levels that the algorithm searches until reaches its target: a black pixel, located somewhere on the map.

[29] `http://www.comp.glam.ac.uk/gisruk/dbk.html`

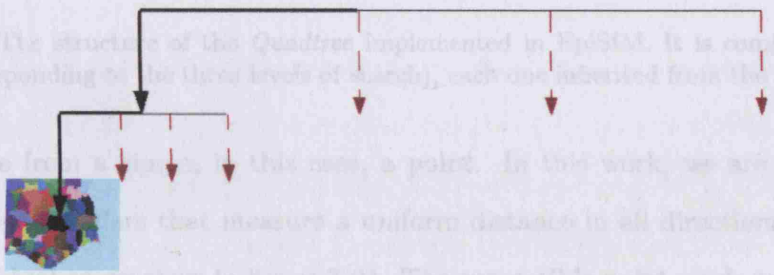


Figure 3-22: Search procedure through the implemented *Quadtree*.

It is now pertinent to look at a different types of operations. When the GIS manipulates the existing attributes of the information, in order to create a classification more appropriate to the questions being asked, we call *Reclassification* (DeMers, 2000). There is a method of *Reclassification* which is frequently called by the infection model and that is *Buffering*. A *Buffer* is a polygon created through reclassification at a spec-

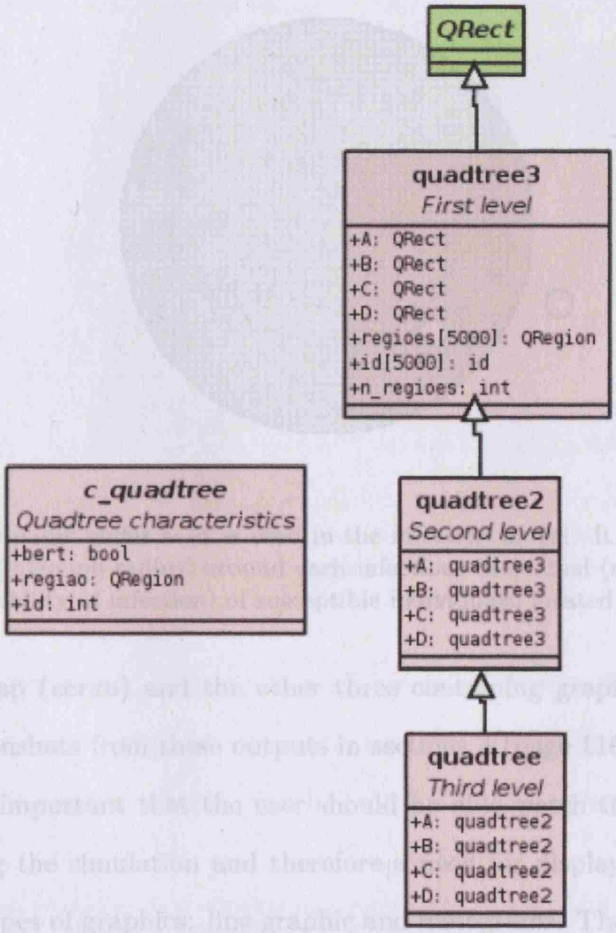


Figure 3-23: The structure of the *Quadtree* implemented in EpiSIM. It is composed of three classes (corresponding to the three levels of search), each one inherited from the level above.

ified distance from a shape, in this case, a point. In this work, we are interested in generating point buffers that measure a uniform distance in all directions from an infective individual as we show in figure 3-24. The susceptible point needs to be tested to know whether it falls inside the buffer (through a point-in-polygon algorithm) in which case it might be infected.

Finally it is worth describing in more detail the display of geographic information and user interaction, through the UI. This is mainly concerned with the output of the program but not exclusively, for we will see, the user is also able to introduce geographic information into the program. As it was already described in section 3.3 (page 81), the display of information is organized through three dialogs: one with the object that

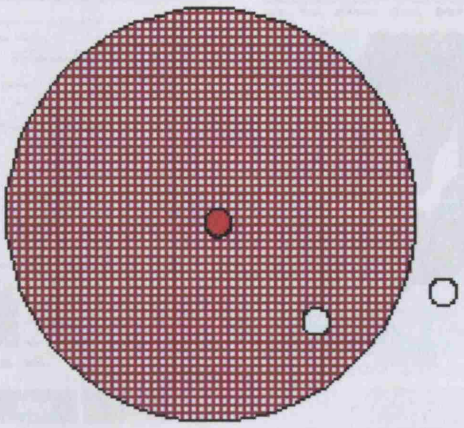


Figure 3-24: The point buffer as it is used in the infection model. It is created a buffer with a certain radius (infection radius) around each infectious individual (red) that determines the infection (or possibility of infection) of susceptible individuals, located within it.

contains the map (*ecran*) and the other three containing graphics. There is a large number of screenshots from these outputs in sections 4 (page 116) and 5 (page 160). It is perceived as important that the user should be able watch the evolution of certain variables during the simulation and therefore a need for display and graphics. There are two main types of graphics: line graphic and histograms. The line graphics concern the evolution of the four epidemic types and epidemic size; the histograms concern the frequency of contacts (relative and absolute). Screenshots of these graphics are shown previously in figure 3-8. The map control allows the display of the point and polygon information from the *Shapefile*. Zooming is natively implemented in this application. As we see in figure 3-25, it is possible to zoom in by drawing a rectangle enclosing the area of interest. As soon as the user releases the mouse, the program will adjust the selected area to the shape of the map container. A right click anywhere on the map will zoom out to the original view.

Besides viewing the data, the user can also input some geographic information to the model, through the map. After clicking on the "Generate Infectious Individuals" toggle button at the right panel of the interface, the program switches to input mode and the user can click on the map generating an infectious individual at that specific

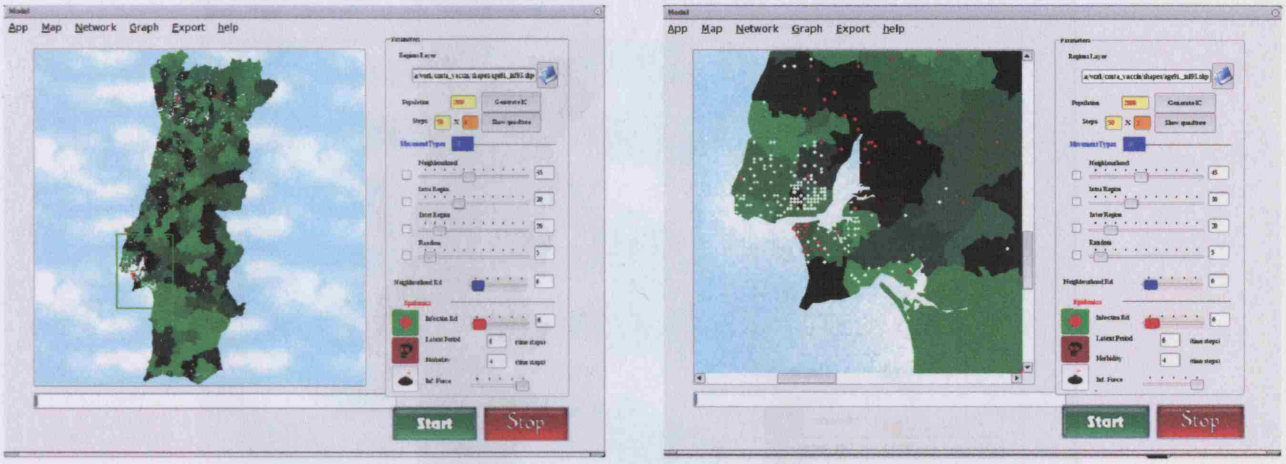


Figure 3-25: Example of the zooming functionality in EpiSIM. In the first screenshot (on the left), the user draws a rectangle enclosing the area of interest and in the second screenshot (on the right), the map is zoomed to fit that rectangle.

location. In the same way, clicking on the "Generate Infection Barriers" toggle button will activate the input mode for vaccination areas. The program will create a buffer around the point location chosen. These two functionalities are illustrated in figures 3-26 and 3-27.

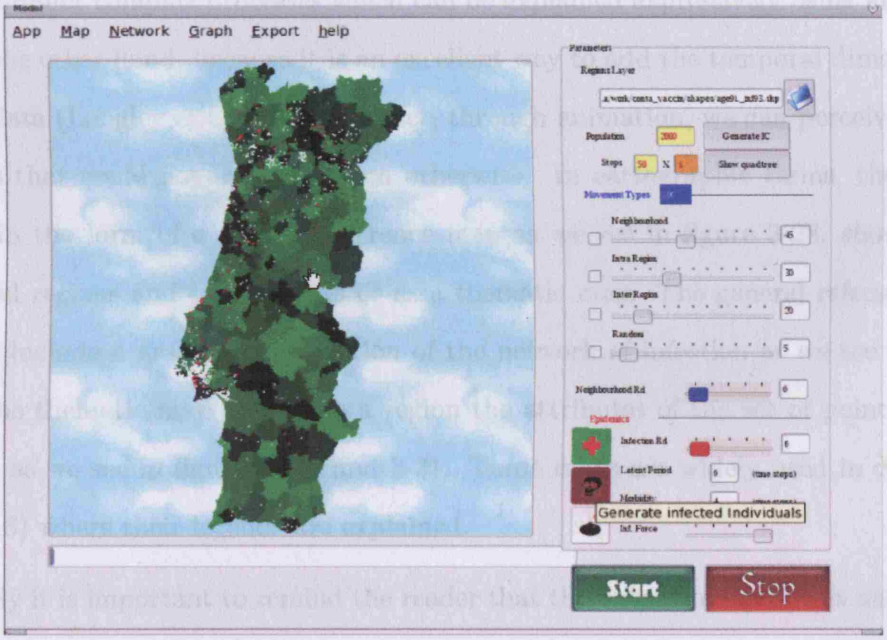


Figure 3-26: Illustrations of the input of infectious individuals through the user interface. The hand cursor points to the place where the user is introducing one infectious individual.

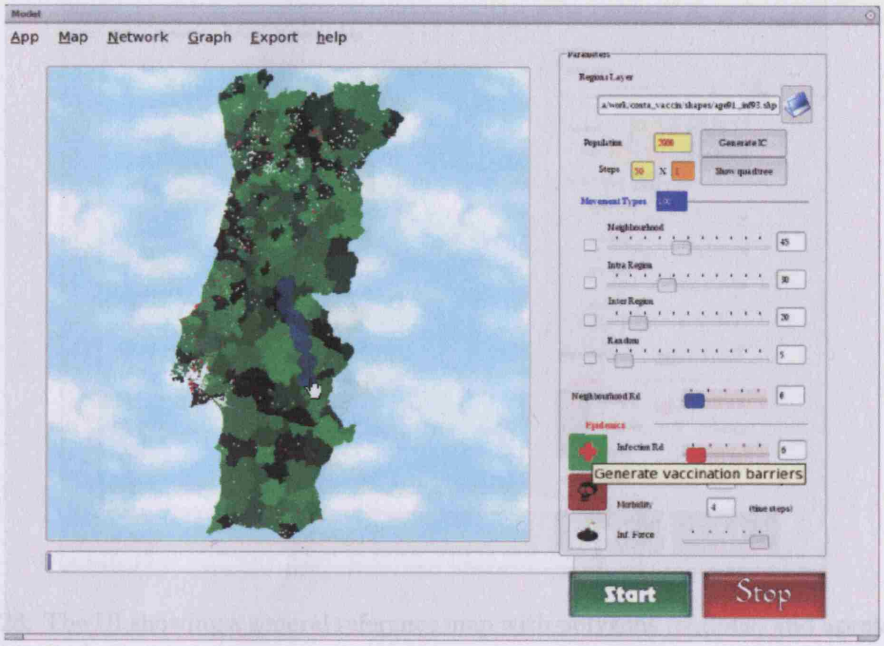


Figure 3-27: Illustration of the input vaccination buffers through the user interface. The buffers are represented in blue.

During the simulation process the map container switches to an animation with the location and attributes of agents. This tool was developed on the one hand because the maps represent complex processes which can be explained expressively using animation and on the other hand, because it is an excellent way to add the temporal dimension to spatial data (Longley et al., 1999). In fact, through animation, we can perceive spatial patterns that could not easily be seen otherwise. In cartographic terms, the output can be in the form of a general reference map as we see in figure 3-28, showing the polygonal regions and point agents or as a thematic map. The general reference map can also include a graph representation of the network of infection as we see in figure 3-29. The thematic map assigns to a region the attributes of the set of points that it contains as we see in figures 3-30 and 3-31. These maps are widely used in chapter 4 (page 116) where their legends are explained.

Finally it is important to remind the reader that the output of EpiSIM is not limited to the display of maps and graphics. As a scientific tool, the generated data has to be stored and analyzed, eventually outside the program. In this perspective, we developed functions for exporting data in the form of text files for raw data and as image files

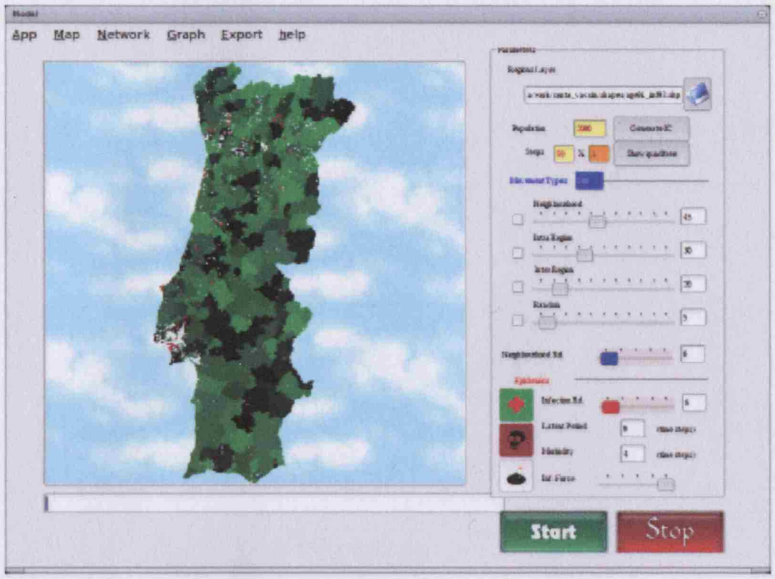


Figure 3-28: The UI showing a general reference map with polygons (regions) and agents, colored according to its internal state.

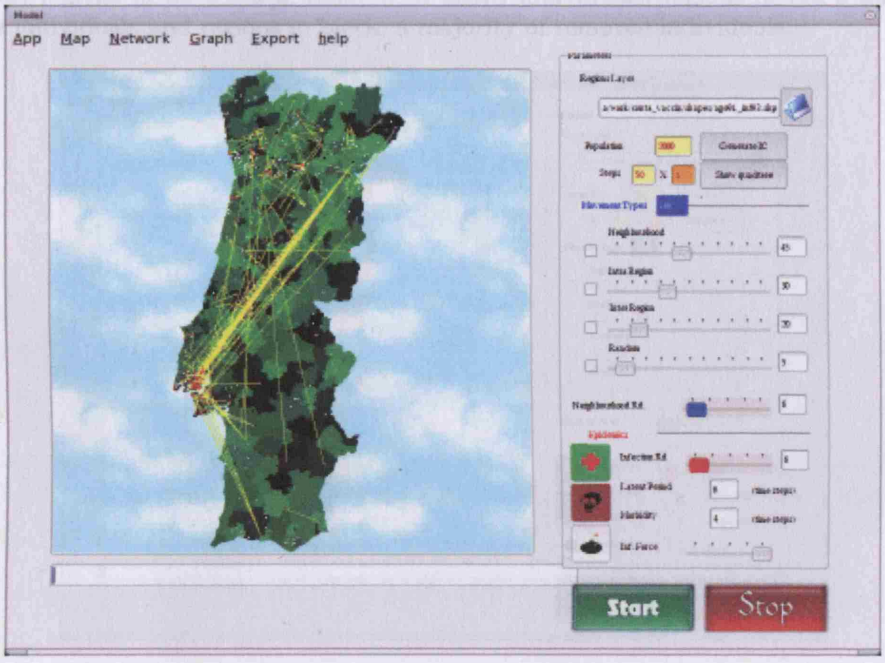


Figure 3-29: A general reference map plus a graph representation of the network of infectious connections.

(with the format *.png*) for maps and graphics.

This section has aimed to provide an insight into the GIS developed for this model. We have developed a tool that rather than just supporting the spatial model, offers display and analysis capabilities in a integrated geographic environment. We believe this

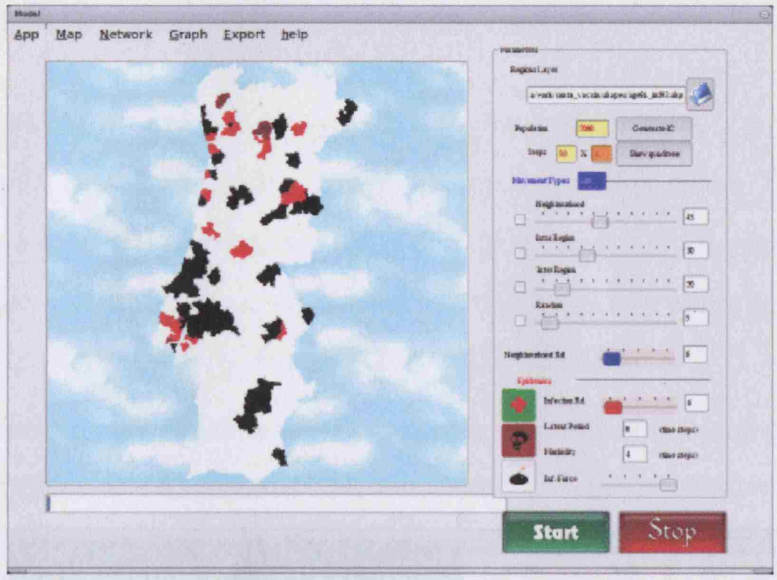


Figure 3-30: A thematic map where an aggregated representation was assigned to each region, according to the dominant epidemic state. Regions in white have a majority of susceptible individuals, regions in dark red a majority of latent individuals, regions in red a majority of infectious individuals and regions in black, a majority of removed individuals.

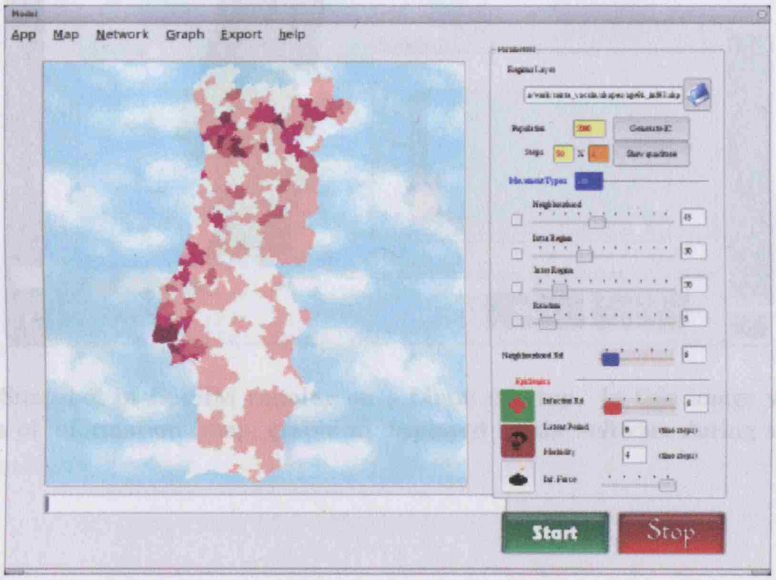


Figure 3-31: A thematic map where the colors represent densities of individuals which were affected by the disease outbreak.

is very relevant in the context of modern spatial epidemiological research, for visualizing the processes of movement and infection and to understand the behavior of spatial epidemics.

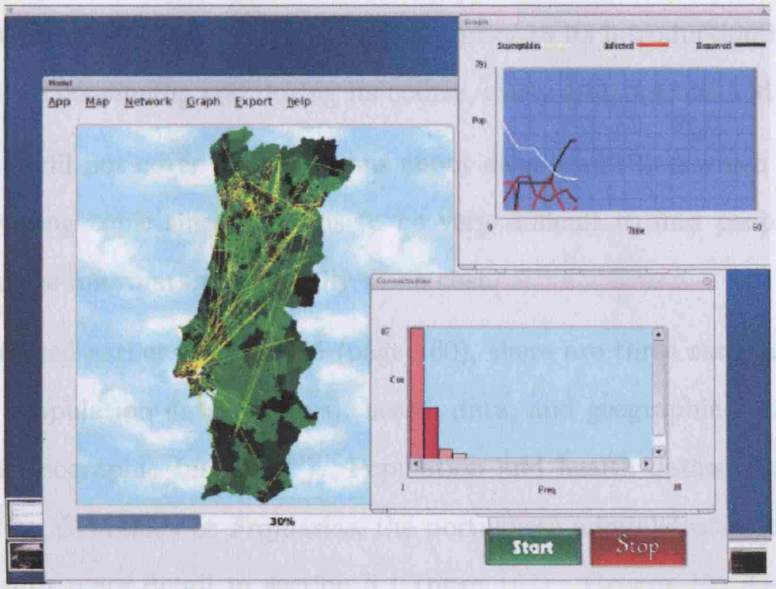


Figure 3-32: Snapshot of EpiSIM running on a Linux desktop. In this image we can see the different types of information (map, graphics) displayed by the software during a simulation.

3.5 Data Preparation and Integration

As mentioned earlier in section 3.4 (page 92), the input of the model is a *Shapefile* whose attribute table contains both demographic and epidemic information. The variables in this table were described in section 5 (page 160). However it is worth saying a few words about the process of compiling, assembling and organizing this information that allows us ultimately to generate the *Shapefile*. This is important, as:

- The dataset was not immediately available to be inserted in the model, and this unfortunately, is often a very common situation in similar studies.
- The whole process of preparing the data, prior to its introduction into the model, was time consuming and during its course, many assumptions had to be made.

This section will not cover considerations about data acquisition which is actually another interesting topic since it seems to be very difficult to find people or agencies, willing to share information, especially at no cost.

As mentioned earlier in section 5 (page 160), there are three categories of data for this study: population data (Census), health data, and geographic data (IGEO - the Portuguese Geographic Institute[30]). Population and health datasets are point data, aggregated by *Concelhos* or *Freguesias*, the portuguese administrative units which are described with more detail in section 5.1 (page 161). Geographic data are polygon *Shapefiles* with *Concelhos* and *Freguesias*. The fact that we are dealing with different scales (*Concelhos* or *Freguesias*) and that these zone arrangements have been modified through the years, introduces the Modifiable Area Unit Problem (MAUP). The level of spatial aggregation and arrangement of these spatial units (zoning) has a marked effect on correlation between the variables (Longley et al., 1999) which may introduce a bias into the study. On the other hand, once point data is aggregated into regions, it is important to bear in mind the *Ecological Fallacy*: relationships between variables observed at one level of aggregation may not hold at a less aggregated level (Longley et al., 1999): in other words, we have to be careful in assigning the characteristics of a

[30] `http://www.igeo.pt`

region to an individual and vice versa which may happen in the case of an agent-based model. However ideal datasets are not available in real world, and we have to deal with the ones we have with the appropriated reserves. This leads us to the next group of issues: accuracy, errors, and uncertainty. These problems are not exclusive to GIS but appear generally in any database. In this study, the data came in different formats, with some of them being extremely inadequate. Sometimes the data fields were not kept the same over the years and there were various errors which required an error check and consistency effort. The first stage of this process was to assemble all this data, that came in the form of excel spreadsheets, text files, and web pages, in a way that the information could be easily accessed and related. This was accomplished through the development of a relational database, which is defined as an organized repository for data (Beynon-Davies, 1992).

There is a diversity of database models but the general success of the relational model can be attributed to the fact that it is so easy to understand, design, and build. Relational databases follow the record-based model which means the database is structured as a collection of files in fixed format records. Although earlier database systems such as those based on hierarchies or networks were also record-based, they proved to be strongly linked to physical details (Beynon-Davies, 1992). A relational database is a collection of tabular relations, each with one with a set of attributes and the data in relation is structured in a set of rows, also called tables (Longley et al., 1999). To summarize, we can say the relational data model is made up of one data structure, eight fundamental operators and two inherent integrity rules (Beynon-Davies, 1992). The Structured Query Language (SQL) has become the standard query language for relational (and non relational) databases. It allows us to define, manipulate, and control data, and it might be used in direct interaction with the database, or embedded in a general programming language. A Database Management System (DBMS) is software that manages access to databases (Bobrowski, 1997). The software chosen in this case was *Oracle*, a powerful object-relational database that offers a set of enhancements such as parallel processing features and advanced security and management settings although

these were not used in this study. However *Oracle* was chosen for being fast, reliable, and easy to work with, which was confirmed during the course of this work. It is important to point out that the latest versions of *Oracle* provide support for spatial data; with *Oracle Spatial*, geographic and location data are managed in a native type within the database which allows the user to perform geographic queries. In this study, although the attribute table of the *Shapefile* is integrated in the database, all the geometry is kept outside, as there is no intention of performing spatial queries but only to prepare input data for the application. In this sense, this is not a spatial database like the hybrid and integrated approaches cited by Longley et al. (1999).

Semantic data models provide an expressive way of representing the information in the database. The Entity-Relationship (also called E-R) data model is commonly used as a conceptual modelling tool for database design. In this model, the world is represented in terms of entities, their attributes, and the relationships between entities. The notation used in E-R diagrams is described on figure 3-33.

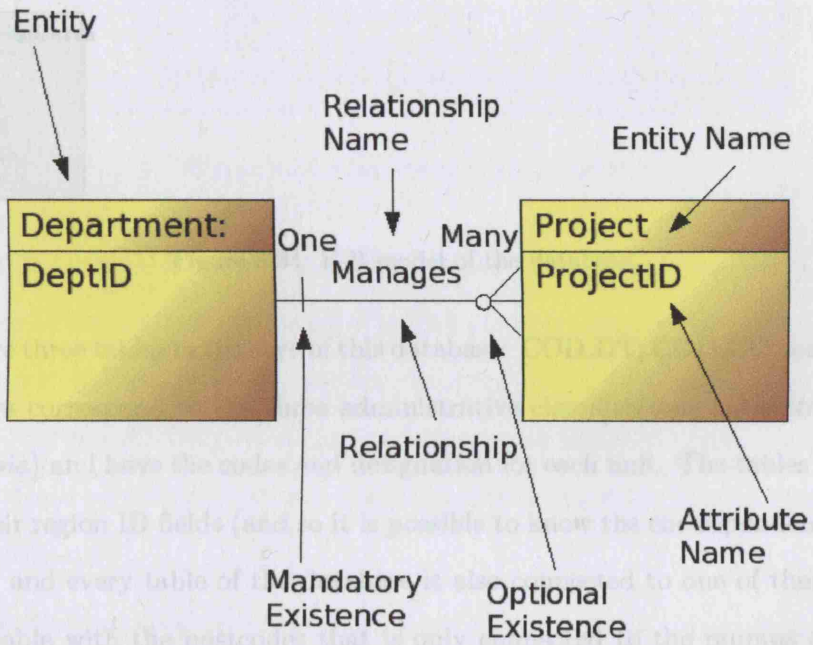


Figure 3-33: Notation on E-R diagrams[31].

[31] based on http://www.utexas.edu/its/windows/database/datamodeling/

Figure 3-34 shows a E-R diagram of the database developed in this project.

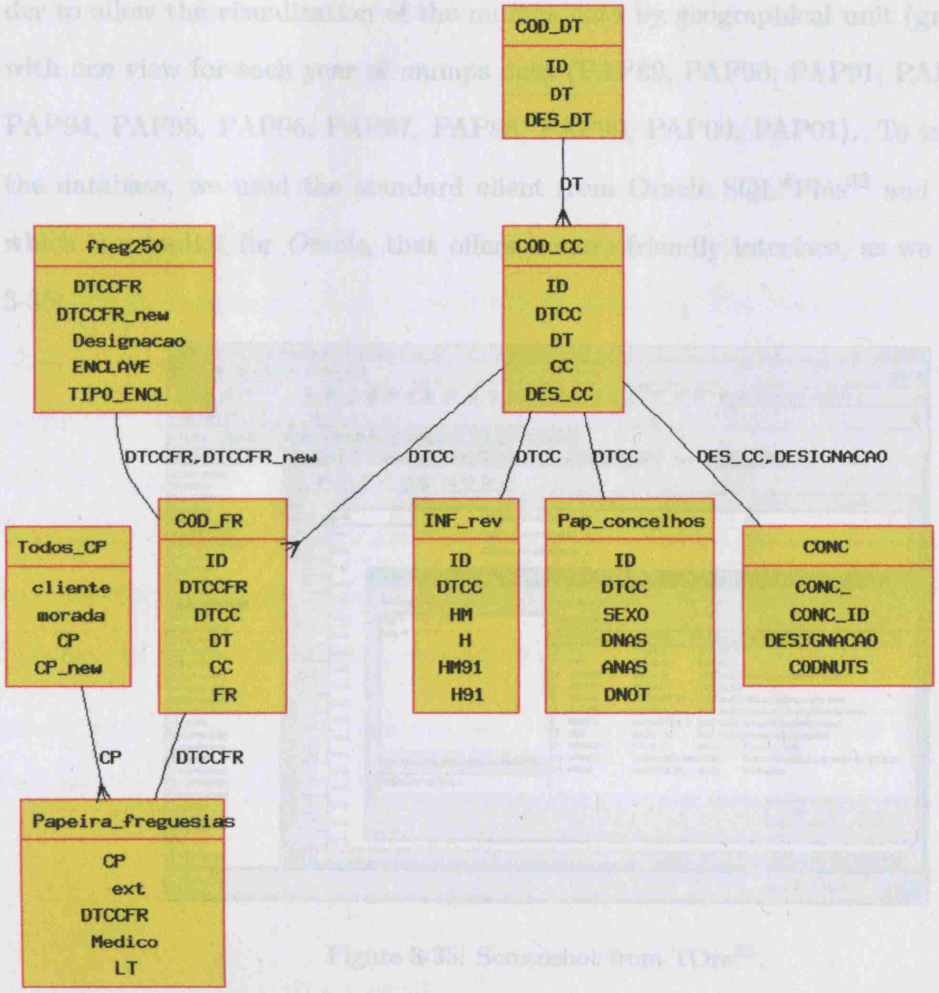


Figure 3-34: E-R model of the database.

There are three tables in the core of this database: COD_DT, COD_CC and COD_FR. These tables correspond to the three administrative classifications (*Distrito*, *Concelho* and *Freguesia*) and have the codes and designation for each unit. The tables are related through their region ID fields (and so it is possible to know the correspondence between each other) and every table of the database is also connected to one of them. Finally, there is a table with the postcodes that is only connected to the mumps data values for 2000 and 2001 (by *Freguesias*) and that has the address correspondents for each postcode. The list of tables is: PAP_CC (mumps by *Concelhos*), PAP_FR (mumps by *Freguesias*), CP (postcodes), INE (population), COD_CC (*Concelhos* codes), COD_FR

(*Freguesias* codes) and COD_DT (*Distritos* codes). Some views were created in order to allow the visualization of the mumps data by geographical unit (group queries) with one view for each year of mumps data (PAP89, PAP90, PAP91, PAP92, PAP93, PAP94, PAP95, PAP96, PAP97, PAP98, PAP99, PAP00, PAP01). To interface with the database, we used the standard client from Oracle SQL*Plus[32] and also TOra[33] which is a toolkit for *Oracle*, that offers a more friendly interface, as we see in figure 3-35.

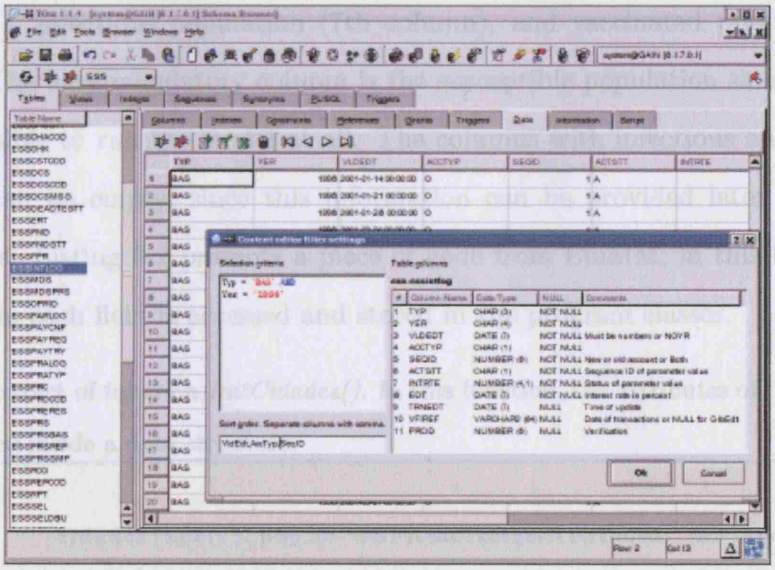


Figure 3-35: Screenshot from TOra[33].

After the database was ready, it was then possible to prepare, in a relatively easy way, the *Shapefile* to be read by the model. At this point, a decision was made concerning the spatial unit of disaggregation and we chose the *Concelhos*, rather than *Freguesias* or *Distritos*, the reasons being that Census information was at *Concelhos* level and that this unit is closer to the concept of a region, stated in the theoretical model in section 2.2.1 (page 51).

The *Shapefile* used was the *Concelhos Shapefile* from IGEO, using all the geometric information (apart from removing duplicated areas). The attribute table on the *dbf* file was replaced by a table generated through a query within the database. This table

[32] http://www.oracle.com/index.html
[33] http://tora.sourceforge.net

has all the information needed as initial conditions. There were some experiments at this stage in order to decide which information to use. As we wanted to capture the outbreak from 1996, after analyzing the maps that are shown on section 5.1 (page 161), we decided to use the infected population from 1993, and the demography from the closest Census (1991); there are more details about this choice of initial conditions in section 5.2 (page 167). The queries generated in the database provide more information, but EpiSIM actually reads only three fields from the *dbf* file: the susceptible population (2nd column), infectious population (7th column), and vaccinated population (14th column)[34]. The only mandatory column is the susceptible population since without it, it is not possible to run the model at all. The columns with infectious and vaccinated population can be empty, since this information can be provided later through the user interface. Listing 3.2 presents a piece of code from EpiSIM; in this function, the information in each field is accessed and stored in the program classes.

Listing 3.2: Extract of function *InitCidades()*. In this function, the attributes of the *dbf* file are read and stored inside a data structure;

```
cidades[npoly].pop_i= DBFReadIntegerAttribute( in2,npoly, 2)/
    ind_num;
pop_total=pop_total+cidades[npoly].pop_i;
// leh os infectados
cidades[npoly].pop_inf=DBFReadIntegerAttribute( in2,npoly, 7);
cidades[npoly].pop_barr=DBFReadIntegerAttribute( in2,npoly, 14);
```

Although at the beginning of this work we did not know exactly which dataset was going to be used in the model, we verified later on that we would only look at a limited amount of information from the database. However, we believe it was important to build this data system, mainly because:

- It was a way of assembling the information and analyzing it, in order to decide what was important for the model.
- It allowed us to track errors and inconsistencies.

[34]The datasets did not provide any information about the latent population and thus that category was not considered in the initial conditions.

- It provides an easy way to generate attribute tables for the model which allowed several experiments.
- It is a flexible structure that facilitates incorporating new information such as future time series.

Chapter 4

Sensitivity Analysis

4.1 Running the Experiments

The model described in section 2 (page 49) and implemented as explained in chapter 3 (page 75) was the object of a sensitivity analysis so that we could understand better its mechanisms. Since this model is split into a movement model and an infection model, they are analyzed separately in section 4.2 (page 117) and section 4.3 (page 137). It makes sense that during the analysis of one model, the other model is kept unchanged with the same parameters, for in this way, we can understand better the interaction between the coefficients of each model. In this analysis, simulations were run using different parameters and combinations of parameters. When testing the value of a certain parameter, we aim to choose values that are representative of its range of variation; for instance, for testing the β coefficient, which is in section 4.3 (page 137), it was adopted the top limit of the interval, the middle of the interval and a very low value. Many other experiments could be run but the purpose of this chapter was to cover essential aspects of the behavior of the model that could help in understanding the results, rather than exhaustively trying all possible combinations of parameters and a large range of values. Although the purpose of these simulations was not to reproduce any real outbreak of the disease it uses a dataset (maps and population distribution) from a mumps outbreak in Portugal, that will be used as a case study in chapter 5 (page 160). This is used simply as an example, and in this chapter there will be no consideration about the data itself. Lastly, it is important to note that all the screenshots and most maps and graphics shown in this chapter, are taken from the application described in chapter 3 (page 75).

4.2 The Movement Model

To understand better the dynamics of contacts generated by the movement model, we built a network in which the nodes are the susceptible and infectious individuals and the arcs, the infectious contacts. Once the infection is passed, the two individuals stay connected until the end of the simulation, which means the two nodes remain attached

by an arc. As referred to in section 2.2.1 (page 51), this design is borrowed from the concept of a relation which exists in social networks and persists in time. In this case, it does not have significance because the sort of relations that allow for the propagation of the disease (physical proximity) do not remain in time, but the construction of this network allows us to look at the clustering of contacts in certain individuals and how they move in space. To understand better the networks generated by this movement model, each component of the movement was simulated in isolation from any other. The default settings used in these simulations are shown in figure 4-1 through the user interface of the model.

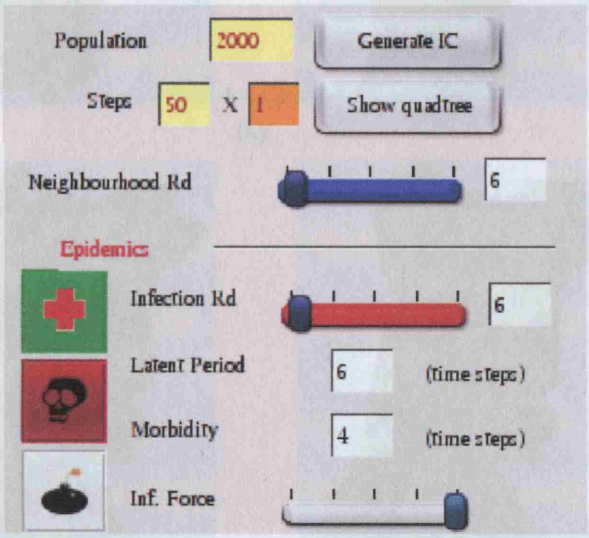


Figure 4-1: Value of the infection parameters used in the sensitivity analysis of the movement model.

In figure 4-2, we show several steps of a simulation which considers only neighborhood movement. In this case, the spread of the disease is very restricted in space as the connections are all short range.

Figure 4-3 shows several steps of a simulation only considering intra-region movement. We can compare these maps with the previous ones and observe how the connections have a slightly larger range and the pattern of distribution of the disease shows a wider spread over the domain.

In the case of a simulation using only inter-region movement, we see in figure 4-4 that the arcs are much longer than in the previous simulations and the clustering of

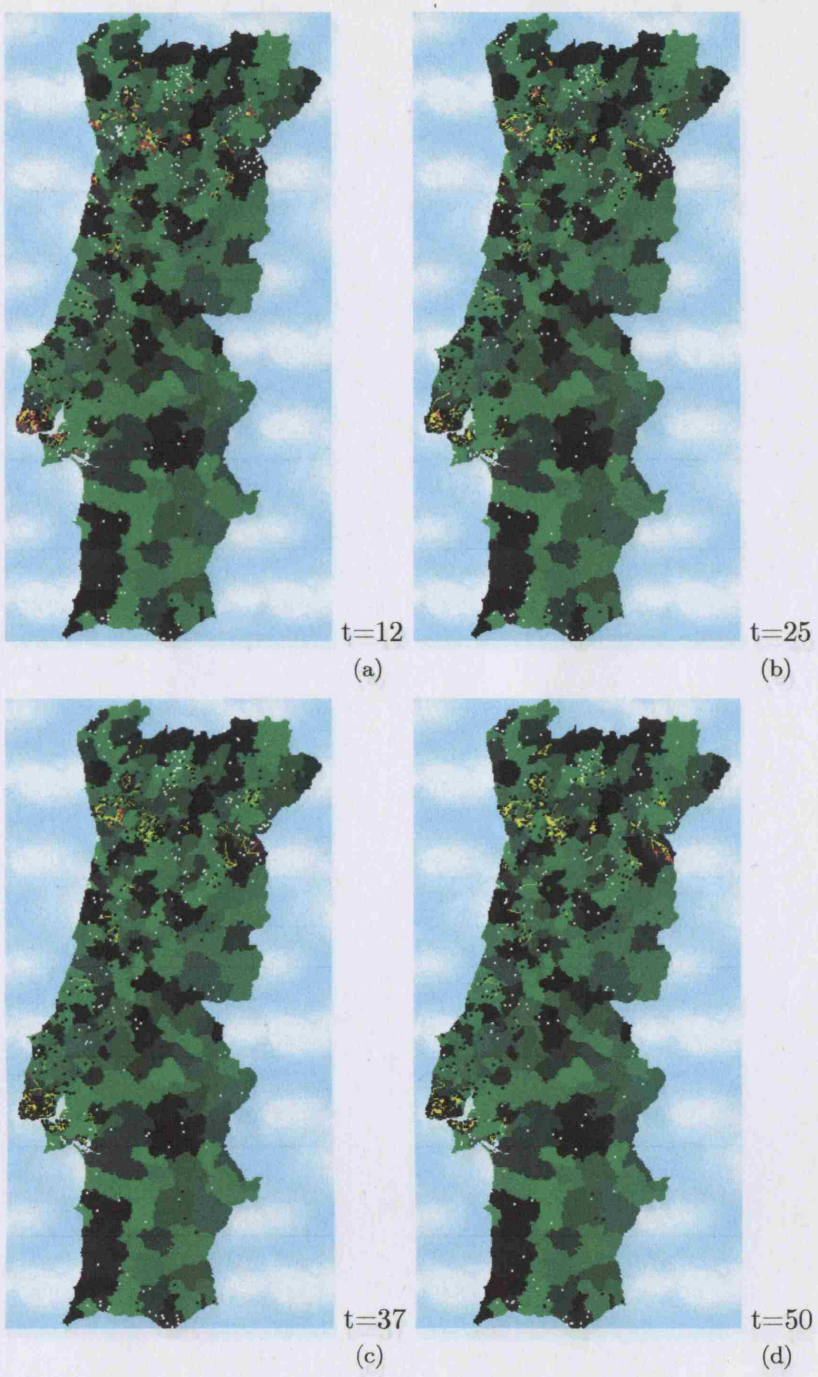


Figure 4-2: Screenshots of the different steps of a simulation considering only neighborhood movement. The regions are represented in different tones of green as a background. The susceptible individuals are represented in white, the latent in dark red, the infectious in light red, and the removed in black. The arcs in yellow represent the infectious connections: the connections between infectious individuals and the susceptibles they infected.

connections coming from highly populated regions (the main cities) becomes visible.

The simulation using random movement, that we show in figure 4-5, shows no orga-

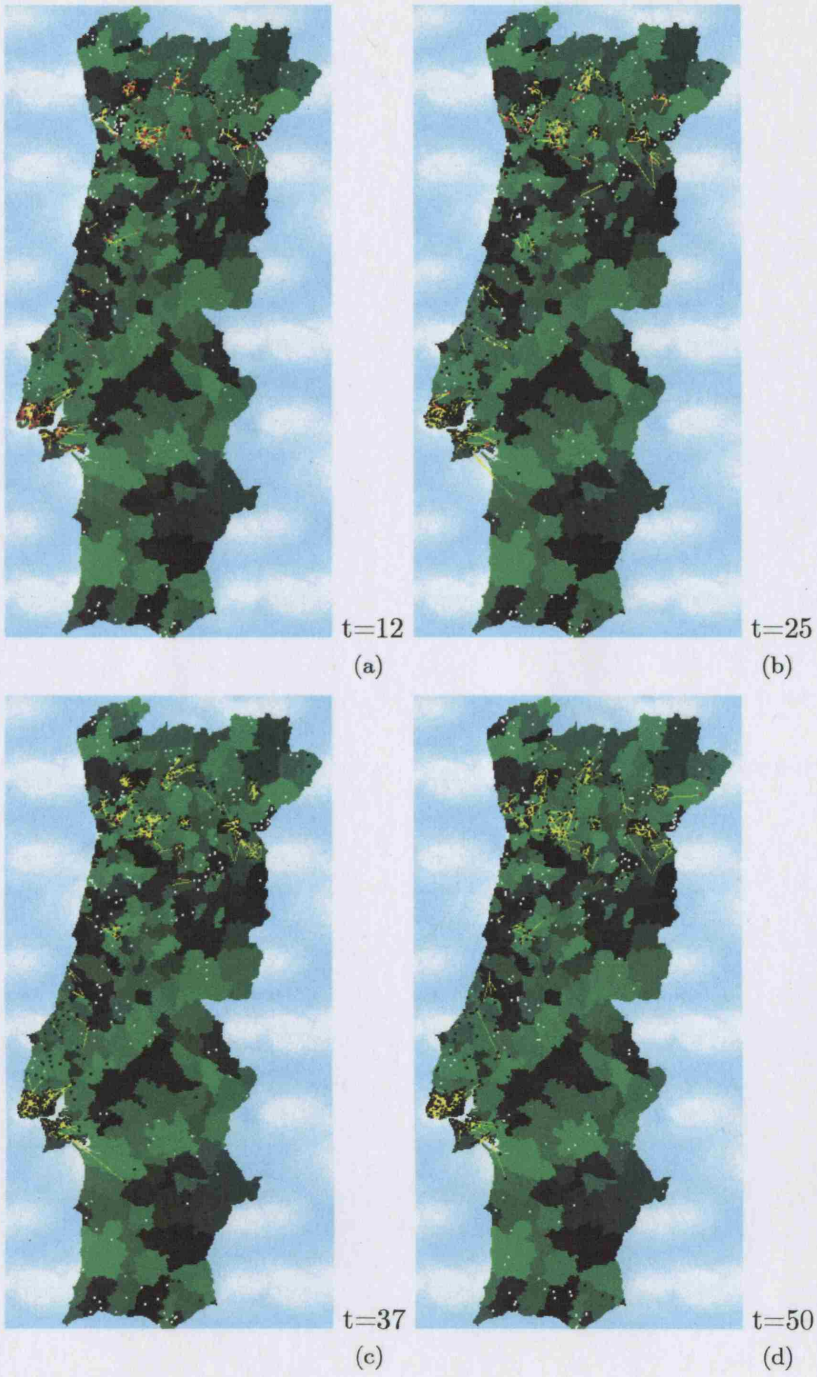


Figure 4-3: Screenshots of the different steps of a simulation considering only intra-region movement.

nized structure and it seems to reproduce a typical random graph.

The charts in figure 4-6 plot the amount of population in each state over time, for the previous simulations in figures 4-2, 4-3, 4-4 and 4-5. The susceptibles series

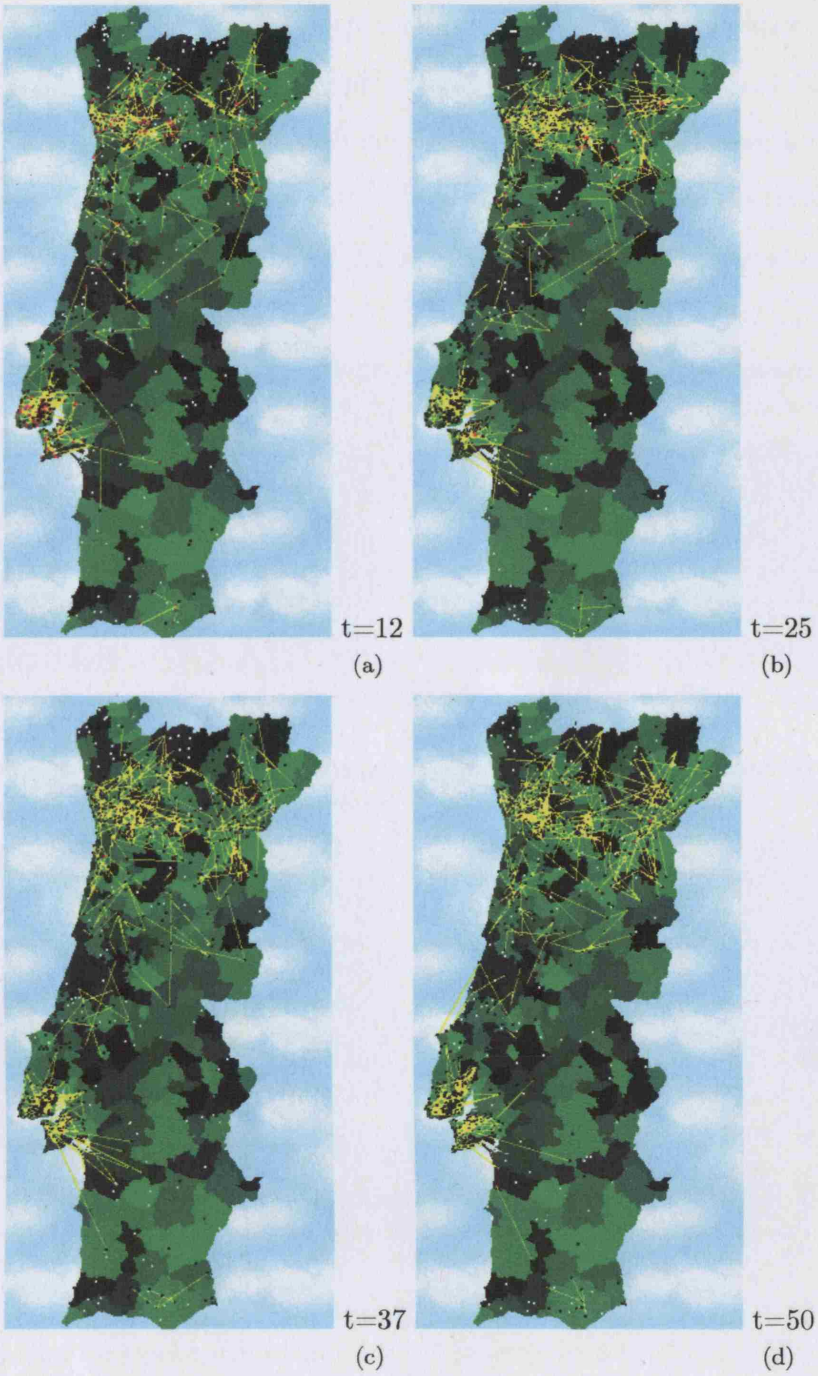


Figure 4-4: Screenshots of the different steps of a simulation considering only inter-region movement.

is represented in white, the infectious series in red and the removed series in black. Although it does not appear in the legend of the user interface because it is a kind of a dummy state, the latent series is also plotted in the graph, in dark red. This chart clearly shows peaks on the latent series which represent the infecting of clusters of

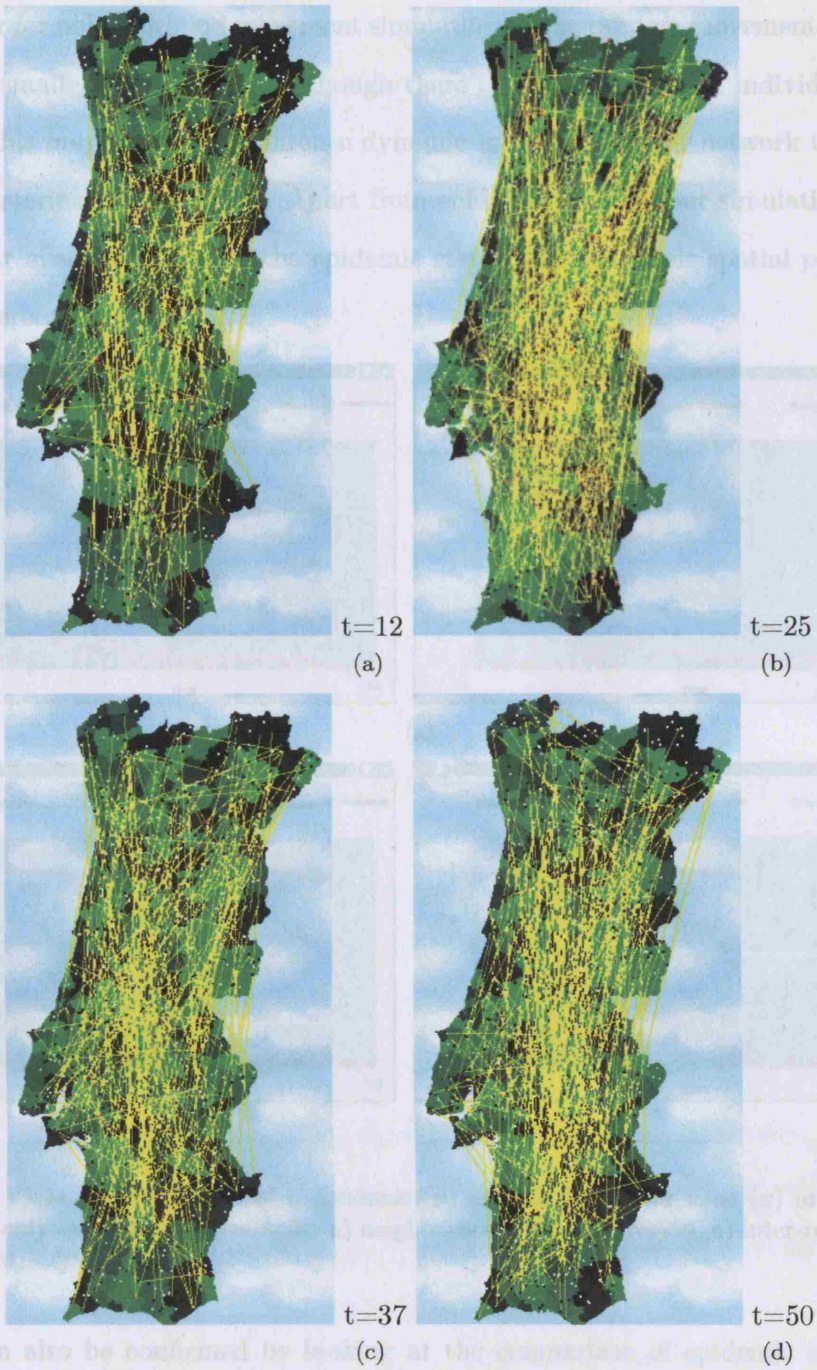


Figure 4-5: Screenshots of the different steps of a simulation considering only random movement.

individuals, and peaks on the infectious series represent the end of the latent period[35]. It can be observed that the epidemic size, the total number individuals affected by the epidemic, increases with the range of movement, being larger for inter-region movement

[35]This subject is more developed on section 4.3 (page 137)

and smaller for neighborhood movement simulations. The random movement simulation presents a smaller epidemic size, although there is a great mixing of individuals in the domain. This may be due to a intense dynamic in the movement network that results in poor clustering of individuals. Apart from epidemic size, all four simulations show a very similar evolution through the epidemic states although their spatial patterns are clearly different.

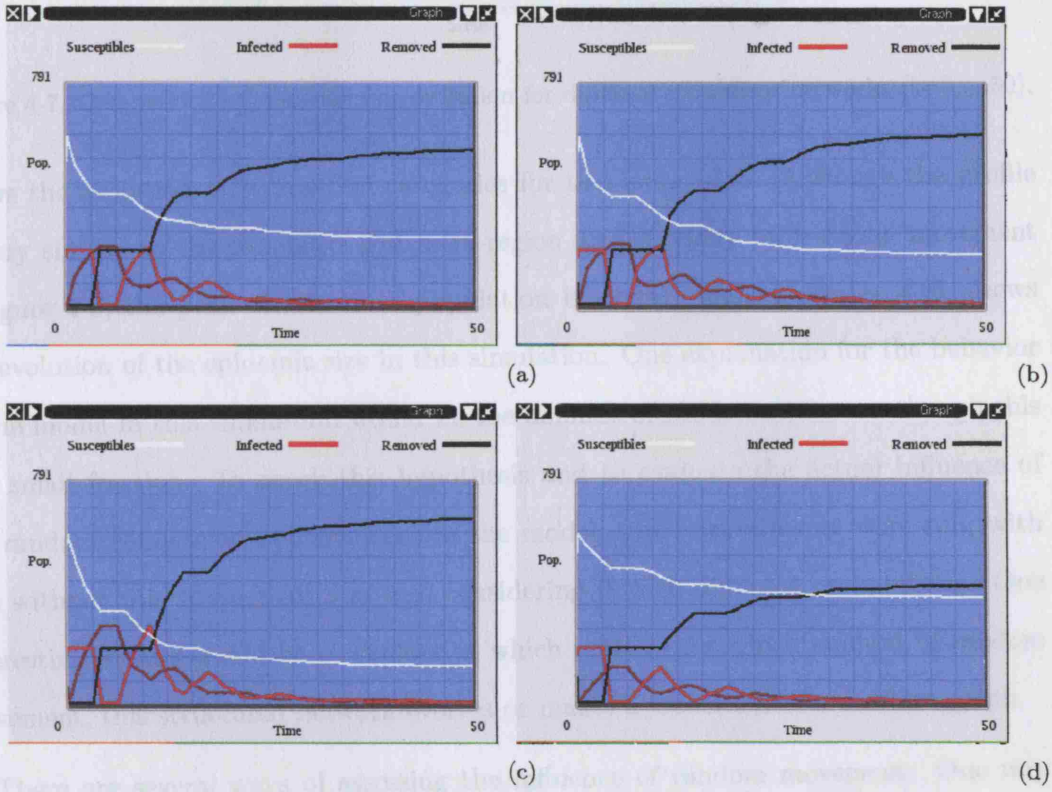


Figure 4-6: Plots of the number of individuals (y) at each state over time (x) in simulations considering only one type of movement: a) neighborhood, b) intra-region, c) inter-region and d) random (t=0,...,50).

This can also be confirmed by looking at the comparison of epidemic size on four simulations in figure 4-7. The higher values correspond to networks with a larger range of movement with the exception of the random network.

Finally in figure 4-8, a simulation is shown which considers the four components of movement in the proportions shown in section 2.2.1 (page 51). The different ranges of movement combined in the contacts network with the emergence of highly populated areas are visible as key points in the spread of the disease. The chart in figure 4-9a

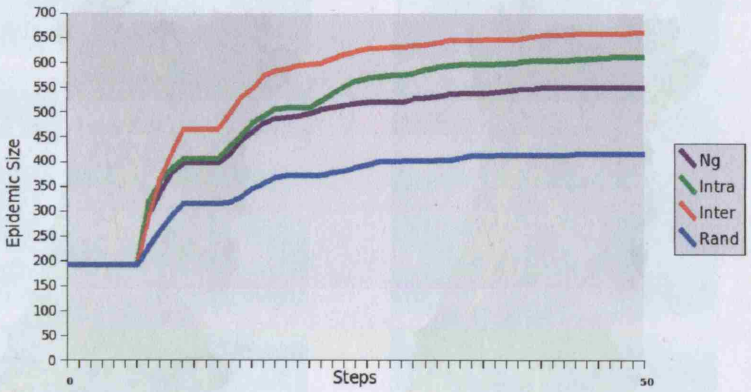


Figure 4-7: Comparison of epidemic size evolution for different movement networks (t=0,...,50).

shows the evolution of population categories for this simulation. Although the profile is very similar to the simulations of intra-region and specially inter-region movement in figure 4-6, the peak of infectious population is slightly higher. Figure 4-9b shows the evolution of the epidemic size in this simulation. One explanation for the behavior of the model in this simulation would be the amount of randomness, even though this is a small fraction. To check this hypothesis and to evaluate the actual influence of the random component of movement in the model, some experiments were run, with and without this component and also considering it with different probabilities. One interesting feature would be to explore at which point or for which amount of random movement, this structured network evolves or makes a transition to a random graph.

There are several ways of assessing the influence of random movement. One way would be to consider only random movement and one other component such as neighborhood or inter-region movement, and vary the proportions between them. Initially, this approach was followed; however, if the purpose of this experience is to evaluate the perturbation introduced by the random component in the network of movement, it is better to consider the full spectrum of components. Since the proportions of random movement changes the structure of the network, it is not possible to keep the same proportions for the other movements in all experiments. This may be an handicap but we attempted to keep the same distance decay logic: greater proportions for the neighborhood movement than for the intra-region and inter-region movements. The

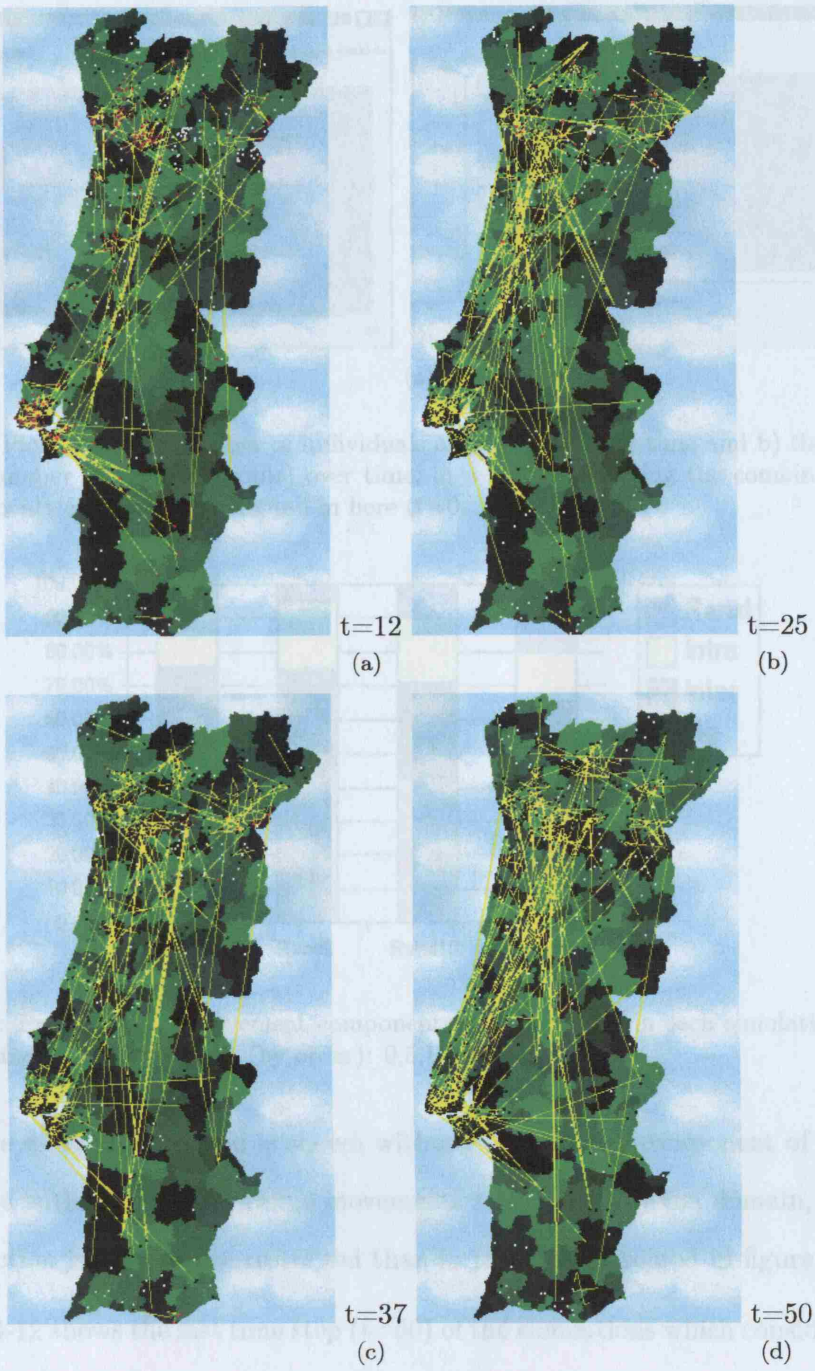
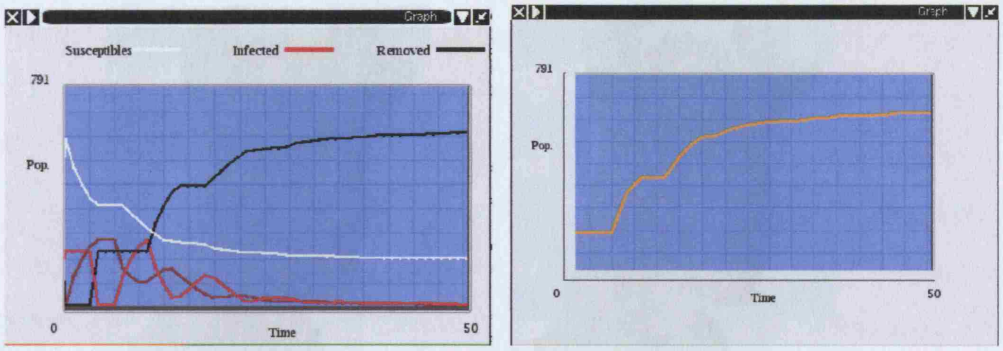


Figure 4-8: Screenshots from a simulation using the combination of the four components of movement adopted here.

random movement varied between zero and an amount that is greater than the inter-region movement (in the last experiment). The exact proportions for these movement components used in the simulations are stated in table 4.1 and figure 4-10.



(a) (b)

Figure 4-9: Plots of a) the number of individuals at each state over time and b) the size of the epidemic (number of affected people) over time, in a simulation using the combination of the four components of movement adopted in here (t=0,...,50).

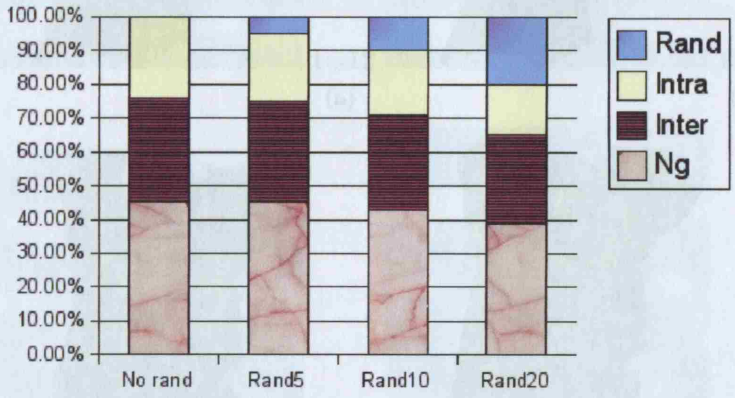


Figure 4-10: Proportions of movement components (y) considered in each simulation (x). The values for random movement are (by order): 0,5,10 and 20.

In figure 4-11, a simulation is shown without the random component of movement. In this case without the long range movements to any part of the domain, the spread of the infection is much more restricted than in the case presented in figure 4-8.

Figure 4-12 shows the last time step (t=50) of the simulations which consider different proportions of random movement. Figure 4-12a corresponds to the example referred to

Table 4.1: Proportions of components of movement used in the random movement assessment simulations.

Sim	*Norand*	*Rand*5	*Rand*10	*Rand*20
Ng	45	45	43	39
Intra	31	30	28	26
Inter	24	20	19	15
Rand	0	5	10	20

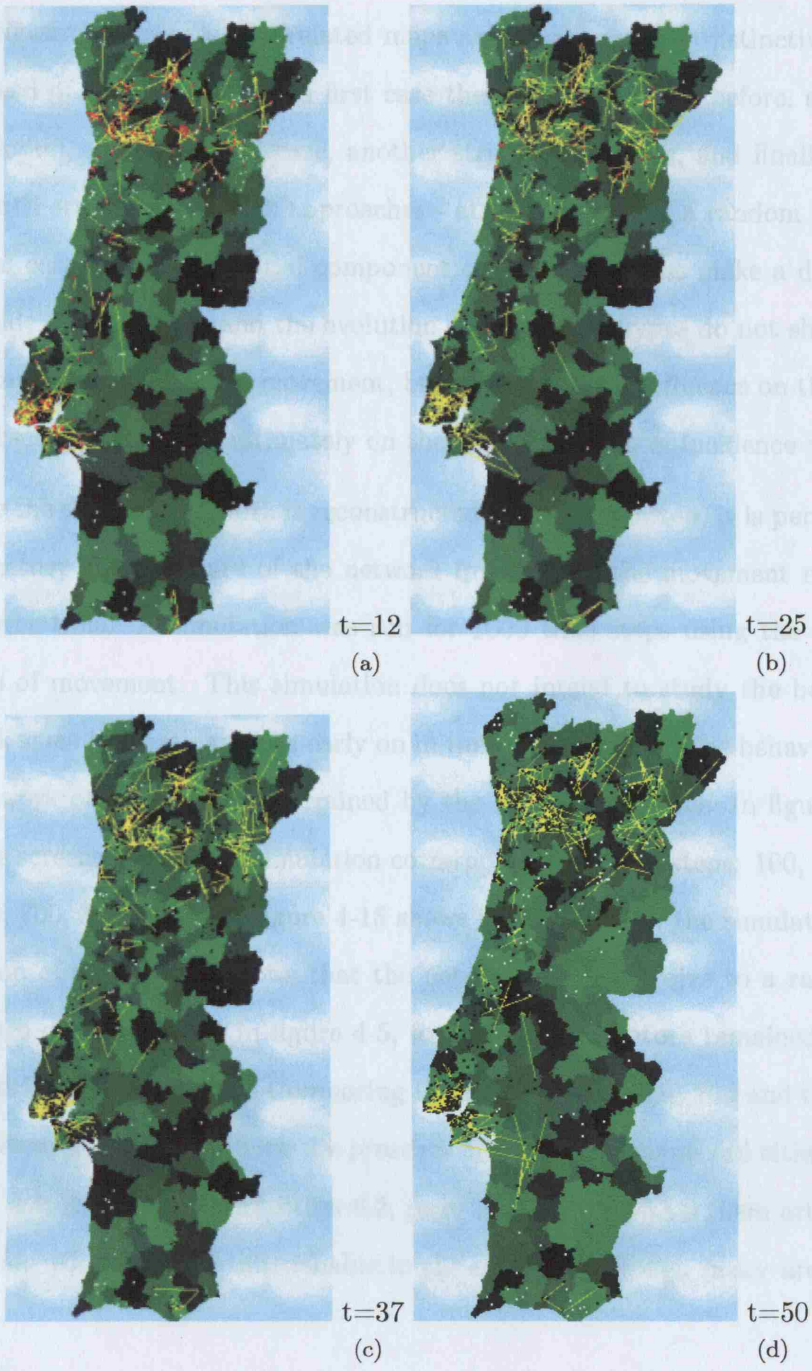


Figure 4-11: Screenshots of a simulation without the random component of movement.

in the previous paragraph where there is no random movement. Figure 4-12b uses the proportions suggested before for each component of movement in figure 4-8. Finally, figures 4-12c and 4-12d show higher proportions of random movement: 10 and 20 %. Although the charts for the evolution of the population states in figure 4-13 do not

show relevant differences, the related maps actually show very distinctive patterns for the spread of the disease; in the first case there is, as was said before, a restriction of the infection. In the second case, another structure emerges, and finally in the third and fourth cases, the network approaches - at least visually - a random graph. This is a typical case where the spatial component of the model does make a difference. The magnitude of the results and the evolution of population types do not show differences in the impact of the random movement, but there is a clear influence on the structure of the contact networks and ultimately on the spatial pattern of incidence of the disease.

Since the dynamic network is reconstructed in each time step, it is pertinent to evaluate whether the structure of the network imposed by the movement model remains stable over time. A simulation was run for 1000 time steps using the standard proportions of movement. This simulation does not intend to study the behavior of the outbreak since it is extinguished early on in the simulation, but the behavior of the contact network of individuals determined by the movement model. In figure 4-14, there are some screenshots of this simulation corresponding to time steps: 100, 200, 300, 400, 500, 600, 700, 800, 900, and figure 4-15 shows the last step of the simulation (t=1000). Inspection of these maps shows that the network did not evolve to a random one, as is the case of the network in figure 4-5, for the main structure remained more or less intact during the simulation. Comparing the maps in time step 100 and time step 1000 this is perceptible through both the presence of the highly connected cities (Lisbon and Oporto) acting as hubs (see section 6.2, page 177); in step 1000 there are still parts of the domain which remain unreachable in the network although many areas have been infected.

Based on the results of this simulation, it is reasonable to conclude that the structure of the network chosen by the movement model remains stable over time. We have not studied the behavior of the model over longer periods of time, more than 1000 time steps, but the time we have used in these experiments is already far longer than the duration of the epidemic, so at least it is possible to state that the network stays stable during the simulation of the epidemic phenomena.

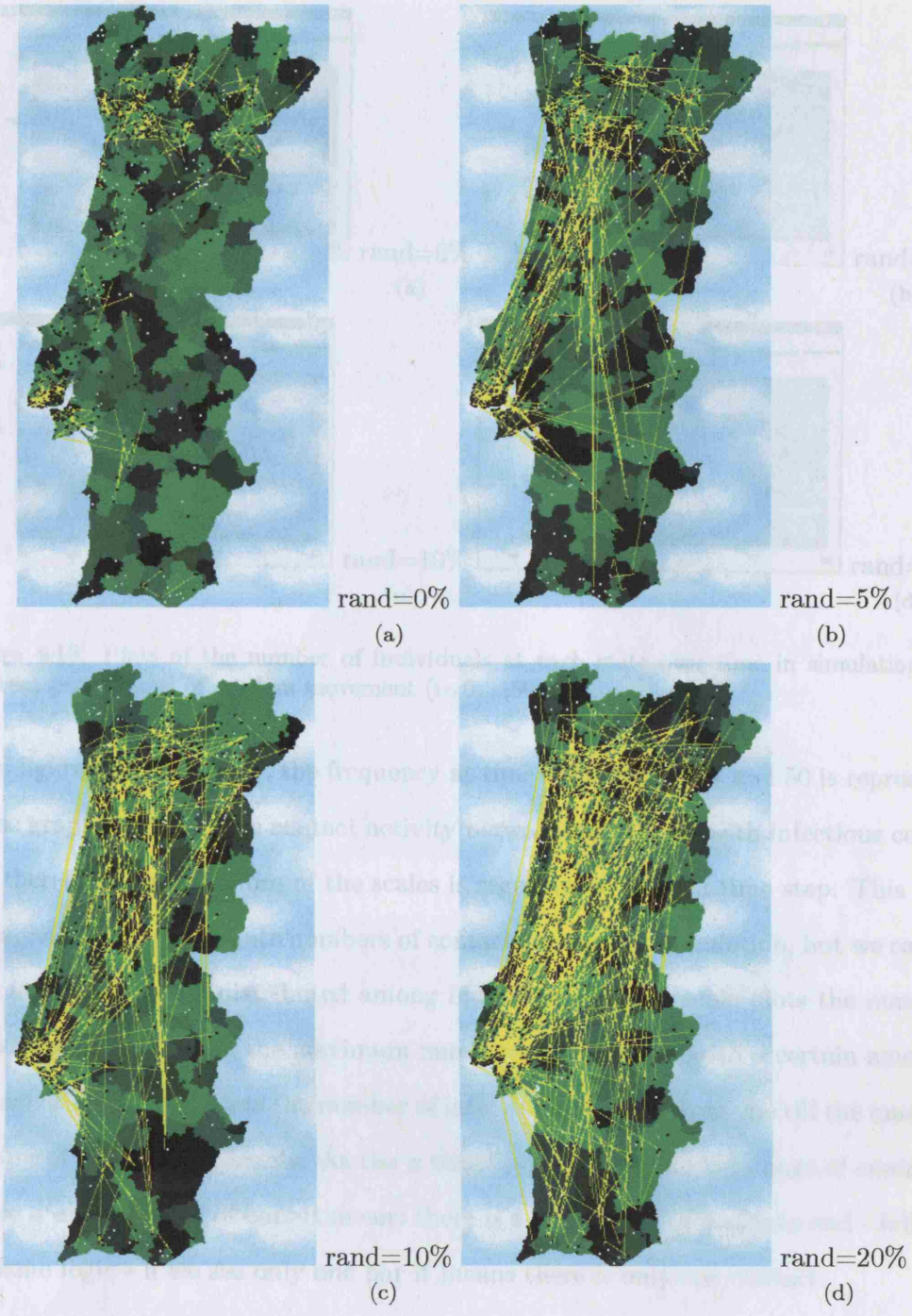


Figure 4-12: Step 50 of simulations considering different proportions of random movement.

Considering each infectious contact as a connection, it is possible to gain some insight into the connectivity of the network by analyzing the frequency of infectious contacts. This is pertinent, once we know there is more evolutionary activity in the most connected nodes (Newman, 1999).

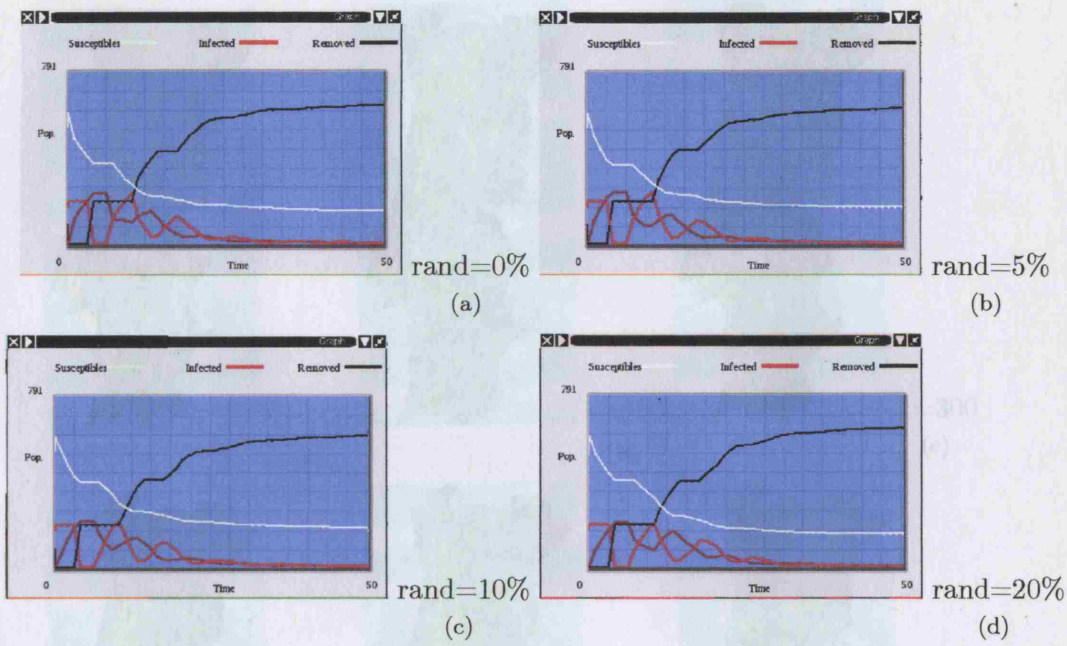


Figure 4-13: Plots of the number of individuals at each state over time in simulations with different proportions of random movement (t=0,...,50).

In figures 4-16 and 4-17, the frequency at time steps 12, 25, 37 and 50 is represented. These graphs compare the contact activity between individuals with infectious contacts and therefore the maximum of the scales is regenerated at each time step. This means we cannot compare absolute numbers of contacts during the simulation, but we can only look at how these are distributed among individuals. The y scale plots the number of individuals from one till the maximum number of individuals with a certain amount of contacts. The x scale plots the number of infectious contacts from one till the maximum number of infectious contacts. As the x scale is divided within the range of contacts, if we see a high number of bars it means there is a wide range of contacts and - following the same logic - if we see only one bar it means there is only one contact.

Figure 4-16 shows the cumulative contacts over the simulation while figure 4-17 shows the instant frequency of connections in the time step. One important conclusion to take from these graphics is that the number of contacts decays exponentially with frequency (number of individuals). There is a small number of individuals with a large number of contacts and a large number of individuals with a small number of contacts, which suggests the power law distribution of connectivities in Kasturirangan's Small Worlds

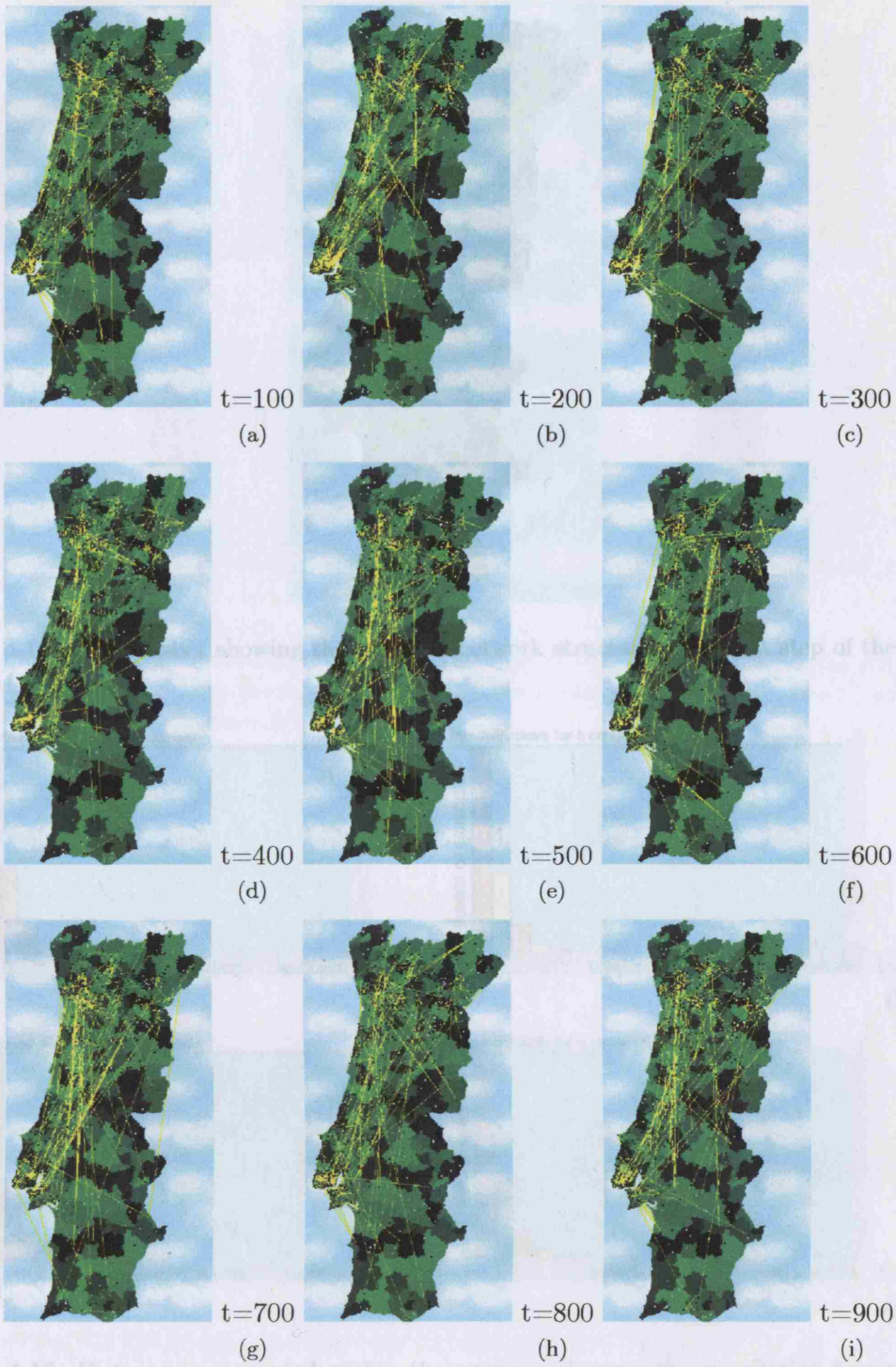


Figure 4-14: Screenshots showing different time steps of a long simulation (1000 steps). In this experiment, we can see how the the infection network behaves, over a long period of time.

model (Newman, 1999) (see section 2.2.1, on page 51). The series corresponding to the last step of the simulation in figure 4-16, was exported using an output function of the program and integrated in a statistical program. Figure 4-18 shows the least squares

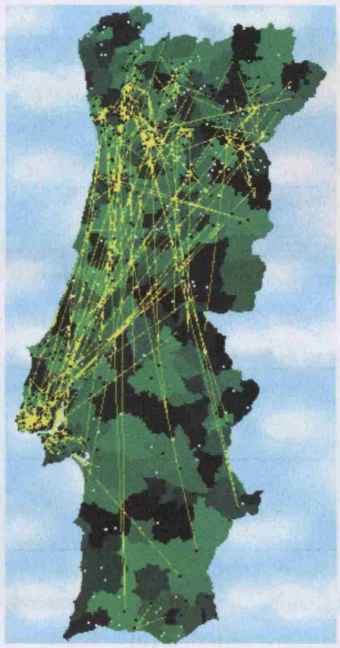


Figure 4-15: A screenshot showing the infection network structure in the last step of the simulation (t=1000).

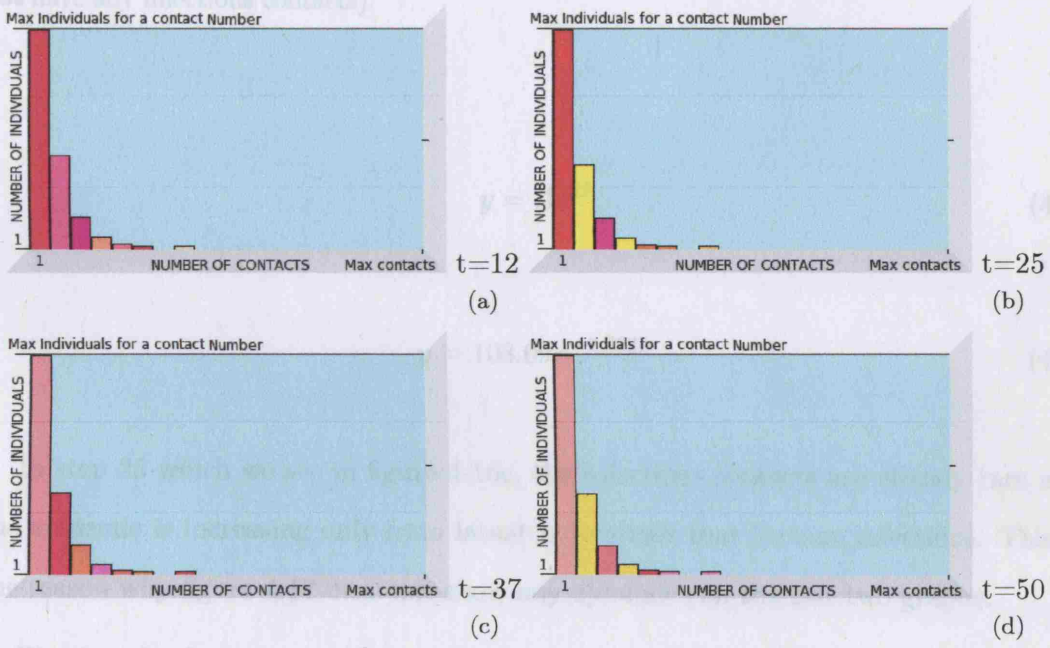


Figure 4-16: Histograms generated within the program showing the cumulative frequency of connections at each time step. The frequency in terms of number of individuals (y) is plotted against the number of infectious contacts (x). Because it is cumulative, the contacts for each individual are added to the contacts from the previous time step.

fitting using a power law distribution, according to equation (4.1), with parameters as stated in equation (4.2).

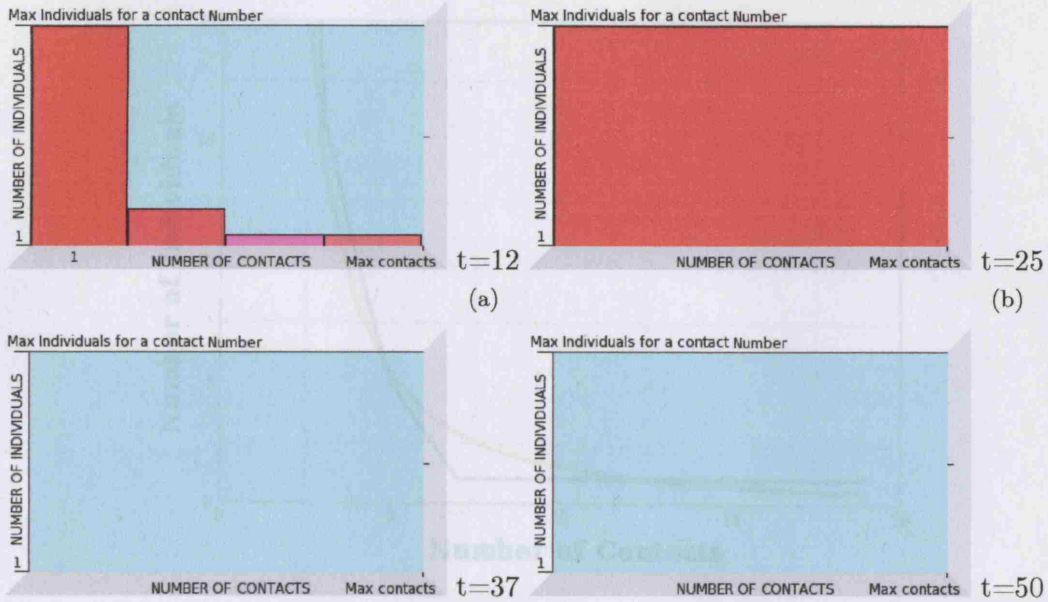


Figure 4-17: Histograms showing the instant frequency of connections at each time step. Since it is instant, we only count the connections that occur on that time step (steps t=37 and 50 do not have any infectious contacts).

$$y = Ax^B \tag{4.1}$$

$$y = 108.09x^{-1.8625} \tag{4.2}$$

In step 25 which we see in figure 4-16c, the infectious contacts are already rare and the epidemic is increasing only from latent individuals that become infectious. This is the reason why figure 4-17 does not show any dynamics on the last two graphs.

Plotting the frequency distributions for simulations which consider exclusively one type of movement can give an idea of how the network shape can determine the connectivities. Figures 4-19, 4-20, 4-21 and 4-22 show the first five time steps for simulations considering only neighborhood, intra movement, inter movement and random movement.

All four simulations reproduce the same profile of decay in the number of contacts

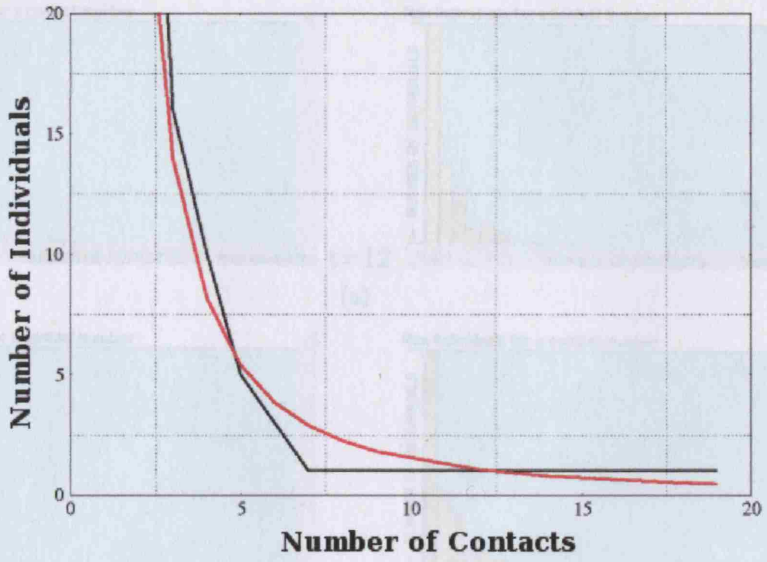


Figure 4-18: This graphic shows the power law fitting of the cumulative distribution of frequencies at the end of a simulation. The distribution of frequencies is represented in black while the power law is represented in red, and it is visible the proximity between the two series. The least squares fitting uses a power law equation with $A = 108.09$ and $B = -1.8625$.

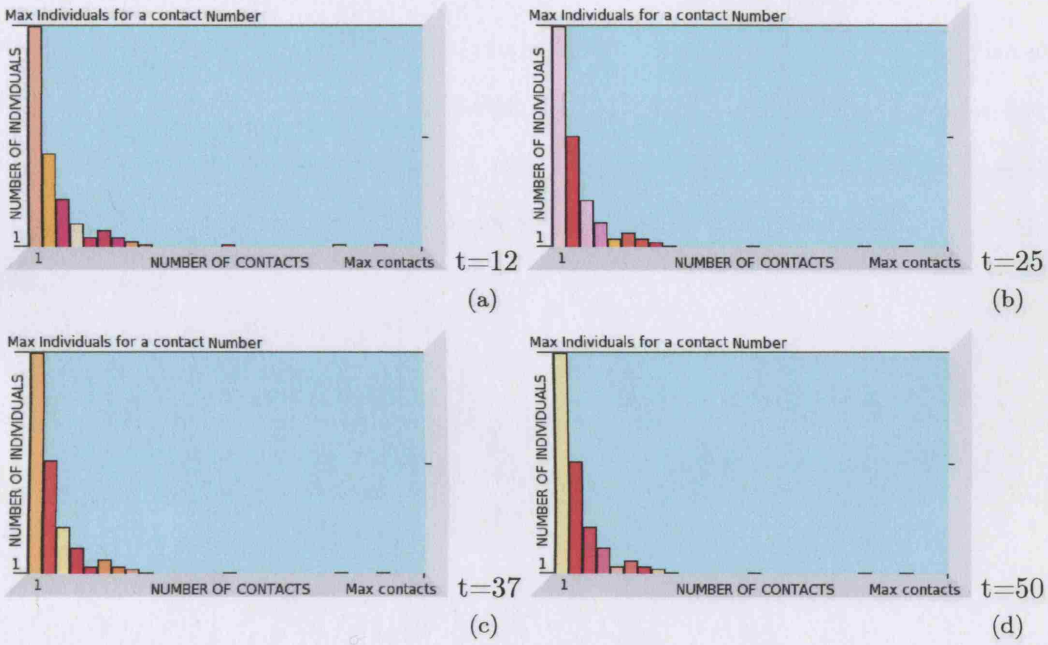


Figure 4-19: Cumulative frequency of contacts in a simulation considering only neighborhood movement.

with frequency: there are more individuals with less contacts. It is also possible to observe, that the shorter range movements produce a broader spectrum of frequencies, and that the random movement is the only one which presents a more peaked distribu-

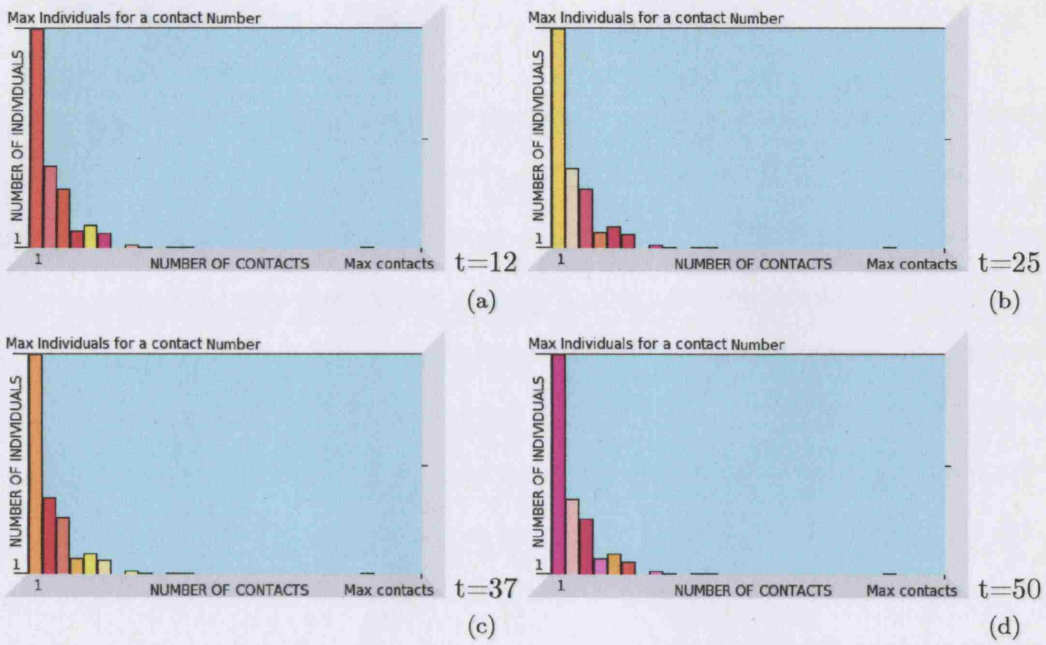


Figure 4-20: Cumulative frequency of contacts in a simulation considering only intra-region movement.

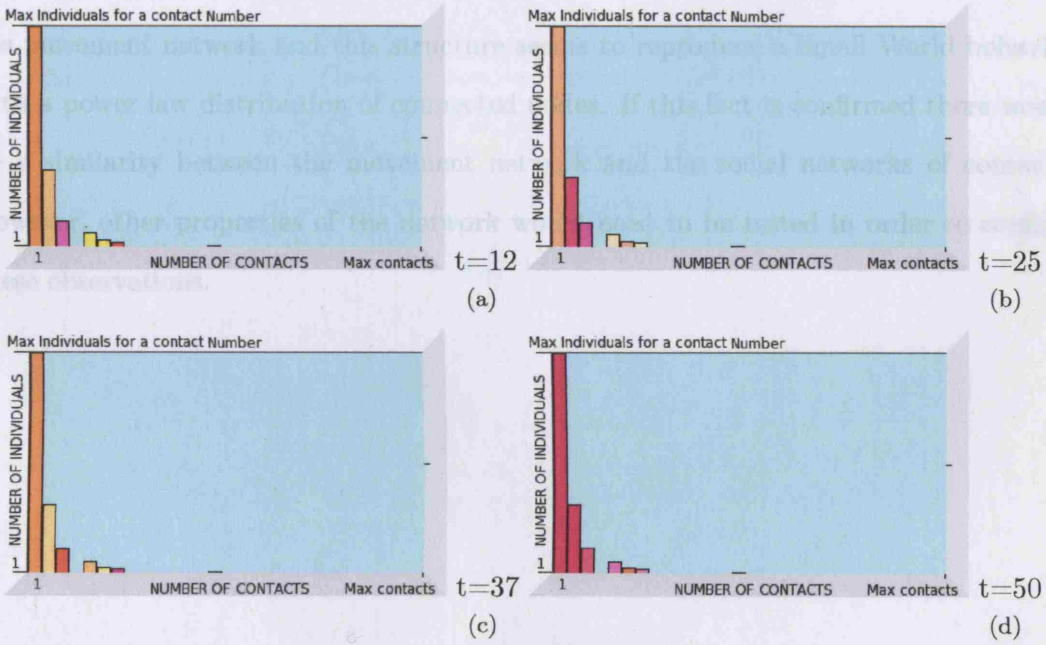


Figure 4-21: Cumulative frequency of contacts in a simulation considering only inter-region movement.

tion with a very small number of individuals with high number of infectious contacts and the rest of the population with practically no contacts.

As a conclusion, we can say that there is a structure of connectivities generated by

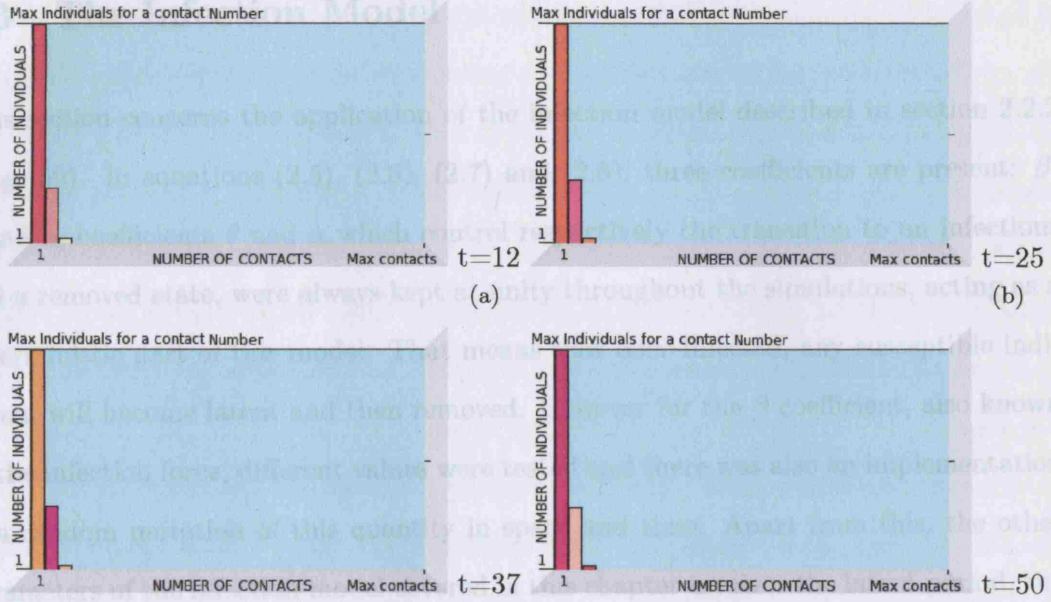


Figure 4-22: Cumulative frequency of contacts in a simulation considering only random movement.

the movement network and this structure seems to reproduce a Small World behavior with a power law distribution of connected nodes. If this fact is confirmed there would be a similarity between the movement network and the social networks of contacts. However, other properties of the network would need to be tested in order to confirm these observations.

4.3 The Infection Model

This section concerns the application of the infection model described in section 2.2.2 (page 59). In equations (2.5), (2.6), (2.7) and (2.8), three coefficients are present: β, θ and α; coefficients θ and α which control respectively the transition to an infectious and a removed state, were always kept at unity throughout the simulations, acting as a deterministic part of this model. That means that once infected, any susceptible individual will become latent and then removed. However for the β coefficient, also known as the infection force, different values were tested and there was also an implementation of a random mutation of this quantity in space and time. Apart from this, the other parameters of the infection model covered in this chapter involve: the latent period, the morbidity, and the infection radius. The parameters of the movement model were kept as presented in figure 4-23 to neutralize their influence in these simulations.

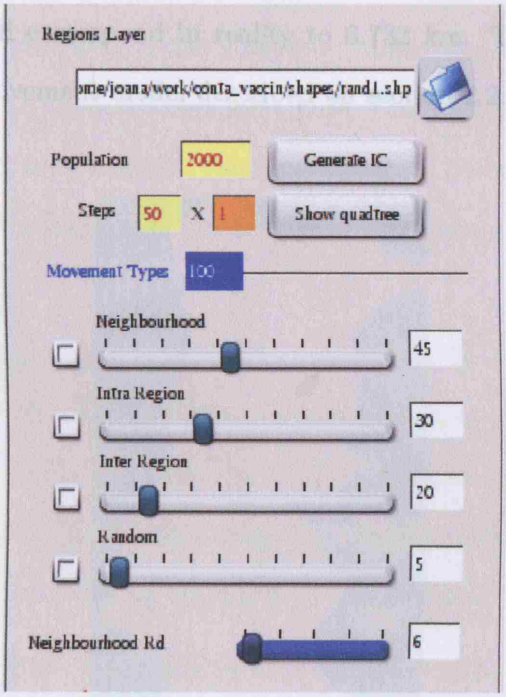


Figure 4-23: Value of the movement parameters used in the sensitivity analysis of the infection model.

The infection radius is a key parameter for a spatial model such as this one. It can be defined as the radius around an infectious individual within which any susceptible individual can become infected. The fact that it is infected or not will depend on the

infection force. The definition of this distance brings into discussion the question of the map scale and map units. The value introduced by the user to the interface will be read in pixels (display units) rather than map units. However it is possible to establish a term of comparison with a real distance and provide an idea of what this value means. When the *Shapefile* is read by the application (see section 3.5, page 109), the map is scaled to fit the viewer and its maximum height and maximum width (the rectangle enclosing the map) are calculated. Once there is information on the *Shapefile* about the real height of the polygon (this is included in the last column of the *dbf*), it is possible to establish a comparison between this distance (the height of the map) and the infection radius introduced by the user. This method is not completely accurate for the *Shapefile* is not projected onto the globe and thus the earth's curvature is not considered in these measures. However considering that the latitude range described by the *Shapefiles* is not large, these errors are relatively insignificant. Therefore an infection radius of 6 pixels for example, will correspond in reality to 6.732 km. The neighborhood radius which is part of the movement model described on section 2.2.1, on page 51, is defined using the same units.

Figure 4-24: Initial conditions for the simulations focused on the infection radius (t=0).

Figure 4-24 presents the initial conditions (t=0) for the experiments. Instead of

the traditional point representation, this representation aggregates results by region although the simulations run at individual level. The relative densities classification assigns a color corresponding to the most representative type in each region. In this way, if there is a majority of susceptibles, the region will be represented in white; if there is a majority of infectious it will represented in red; and if there is a majority of removed then representation is black. A majority of latent individuals will be represented in dark red. In figure 4-25, a simulation is shown using a infection radius of 6, which is the value adopted in the rest of the later simulations in section 5.2 (page 167). As mumps is a disease that spreads through close personal contact (see section 5.1, page 161), the infection radius for mumps must be very short and 6.7 Km (6 pixels) is far too long. However, some simplifications were made in this model (see 5.2 on page 167) when we consider that one individual actually represents a group of individuals, and in this context, it makes sense to adopt a wider radius.

The sequence in figure 4-25 shows how the disease spreads, and then the outbreak is extinguished, once all the infectious individuals reach the removed state. Figure 4-26 presents a simulation with a short infection radius - 1 (around 1.1 Km) - where it is clear that the infection stays restricted in space and dies out very early, not allowing the emergence of a serious outbreak.

In fact, the infection radius is such a key parameter in the model and its influence is so strong that there is no need to run many simulations to come to that conclusion. Figure 4-27 shows a simulation based on a slightly higher infection radius: 22 (around 24.7 Km). The result of this configuration is a pandemic behavior of the disease with the infection reaching all parts of the domain and affecting most of the population.

From these simulations, it is possible to draw some conclusions:

- Due to the spatial spreading mechanism, this model is highly sensitive to the infection radius (as well to the neighborhood radius).
- From these results, it is possible to establish a comparison with real life and to say that the epidemic behavior of a disease is highly related to the proximity

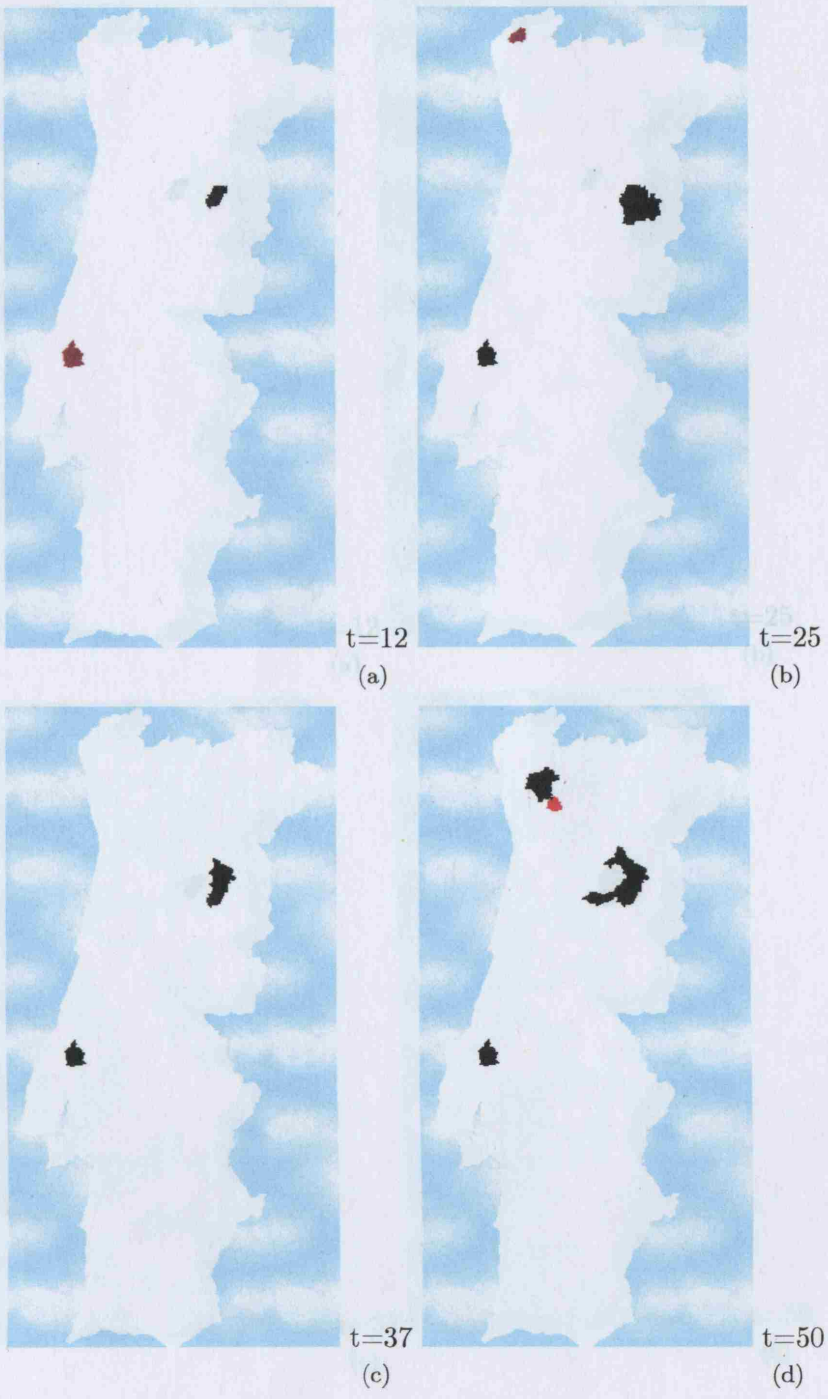


Figure 4-25: Screenshots of a simulation using an infection radius of 6.

it requires for the infection to be passed on. Diseases that do not require close personal contact but can be transmitted through objects or animals (as is the case of vector diseases), have a much wider radius and more serious consequences (as is the case of SARS).

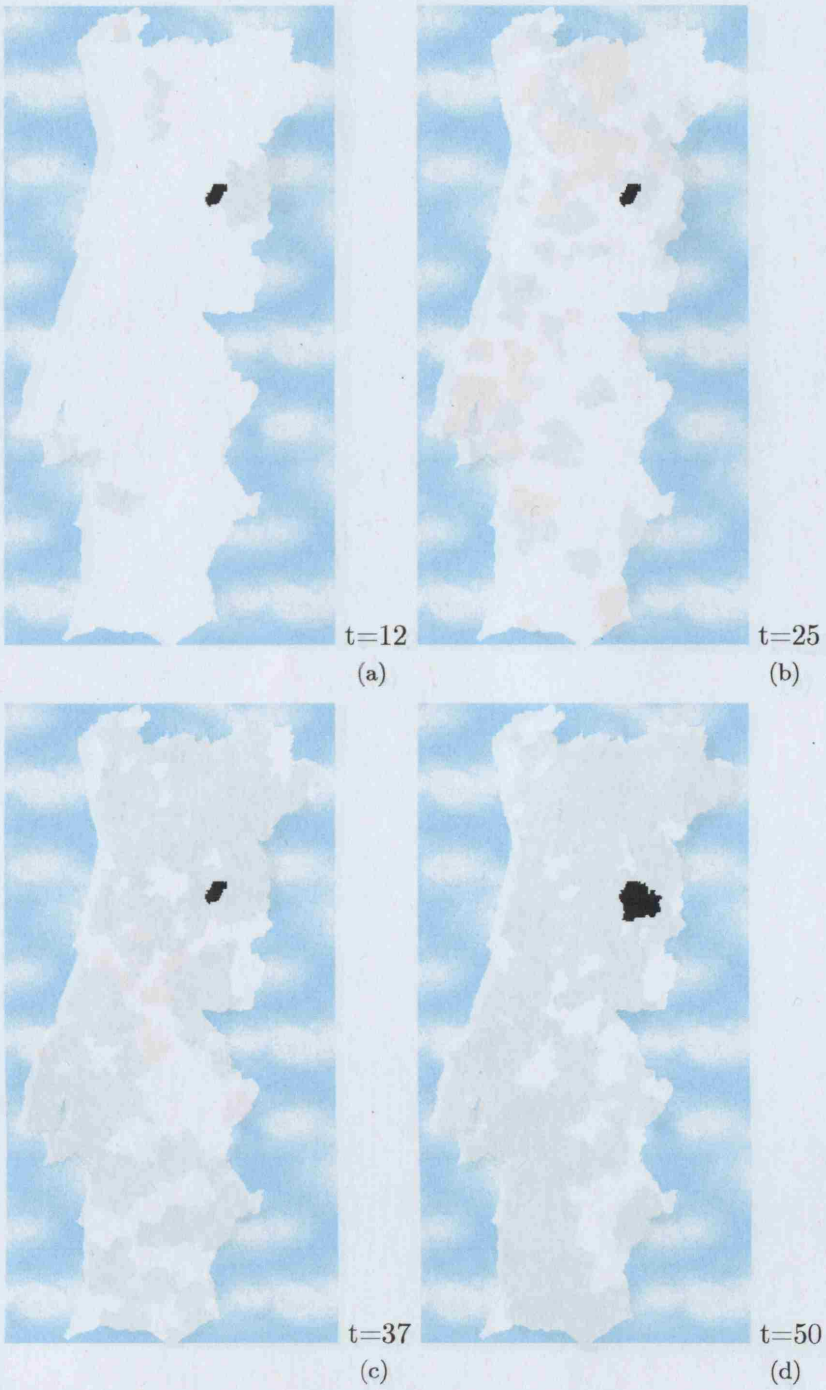


Figure 4-26: Screenshots of a simulation using an infection radius of 1.

The infection force, β, is the coefficient that regulates the infectious contact (section 2.2.2, page 59). Figure 4-28 shows the initial conditions for the simulations which target β. In this case, another type of representation was adopted: each region is classified according to the number of affected individuals and darker colors are assigned to higher

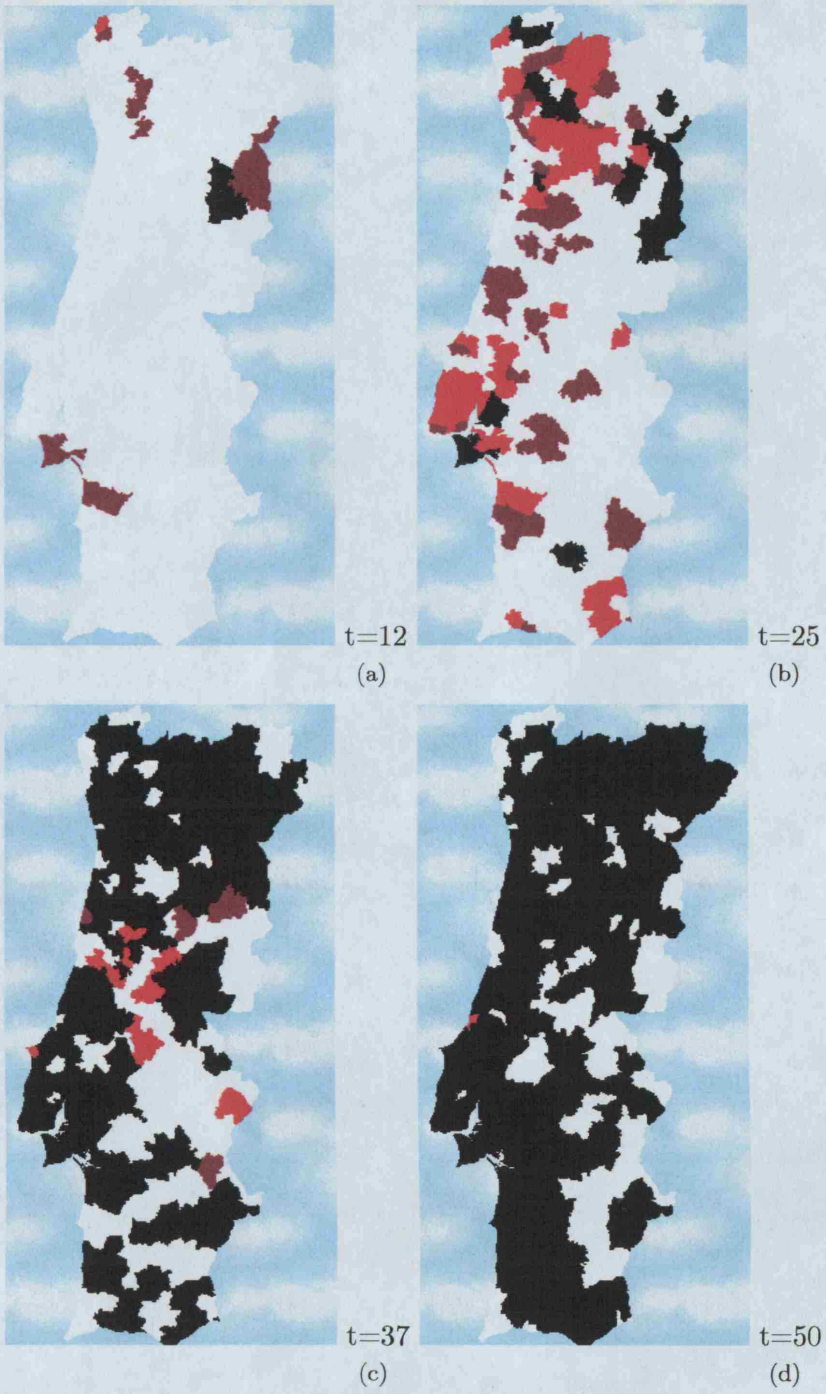


Figure 4-27: Screenshots of a simulation using an infection radius of 22.

epidemic sizes. Exceptionally in these simulations to promote a higher dynamic of infectious contacts, a latent period of 1 and a morbid period of 5 were adopted. The importance of the latent and morbid periods, will deserve some attention in other parts of this section. Figure 4-29 shows the steps of a simulation with β=1. Using a unitary

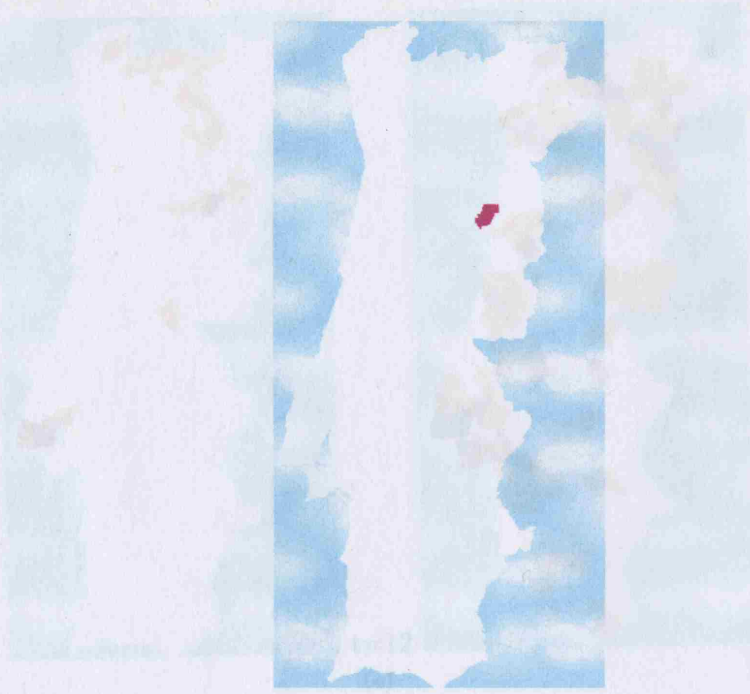


Figure 4-28: Initial conditions for the simulations concerning the infection force (t=0).

β means that all infectious contacts will originate an infection which is a deterministic infection force.

In figure 4-30, a simulation is represented with a small value of β (β=0.25). As would be expected, the spreading of the infection is very restricted, since only a small part of the infectious individuals will spread the disease.

The simulation in figure 4-31 uses a higher β: 0.5. In this case, the infection has more chance of spreading but it is still much less damaging than in the case where β=1. In a real situation for various reasons, infection does not occur every time there is a infectious contact, and so a probabilistic approach would be more appropriate than a deterministic one. However without data, it is difficult to assess the exact infection rate and not much can be done apart from some empirical speculation.

The antibiotic resistance is highly related to the infection force. Ideally, antibiotics entirely eliminate or reduce severely the values of β. However, organisms like bacteria are single-celled creatures with a small number of genes; this means that even a single gene mutation will strongly affect their ability to cause disease[36]. This fact allied to

[36] http://www.niaid.nih.gov/factsheets/antimicro.htm

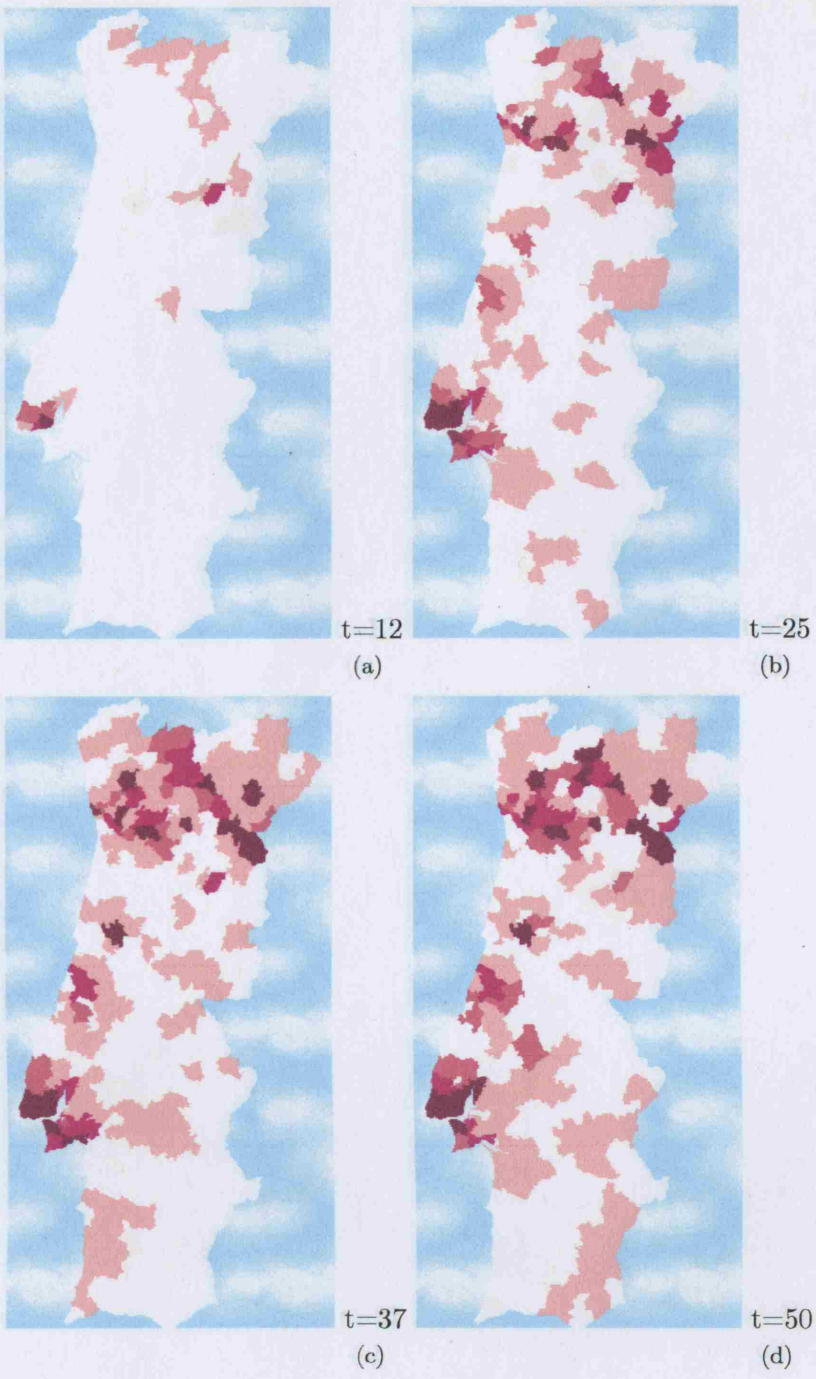


Figure 4-29: Screenshots of a simulation using infection force=1.

their high reproduction rate, provides microbes with the capacity to quickly adapt to new environmental conditions. In order to explore this idea of a dynamic infection force, simulations were run using random mutations of β in space and time which may represent different adaptations of the disease; this simulation is presented in figure

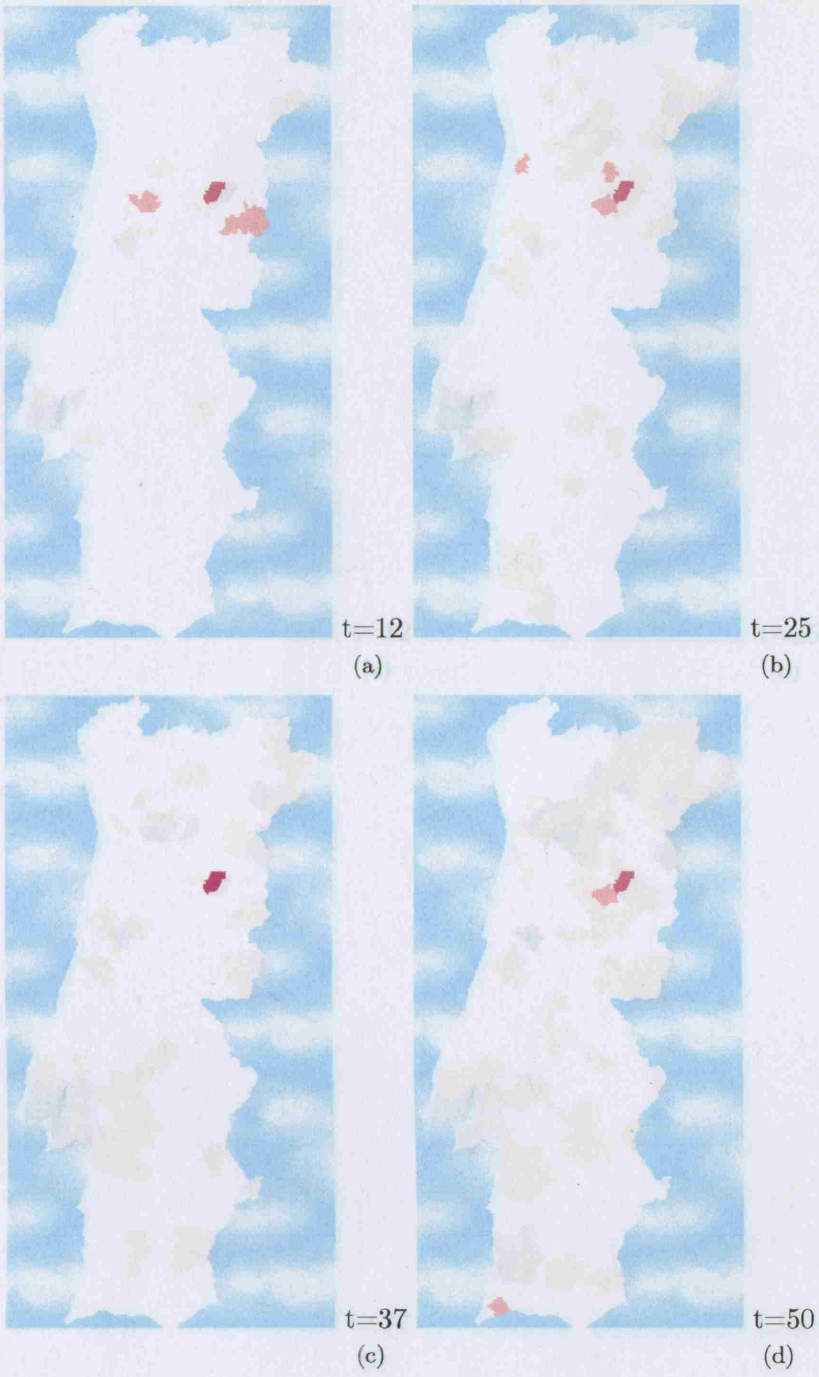


Figure 4-30: Screenshots of a simulation using infection force=0.25.

4-32. In fact, this idea could be taken further, for instance creating scenarios that start with low infection forces and then introducing some increments that would spread quickly over the network. This would represent the situation where a mutation helps a microbe to survive to the exposure to an antibiotic drug and quickly become dominant

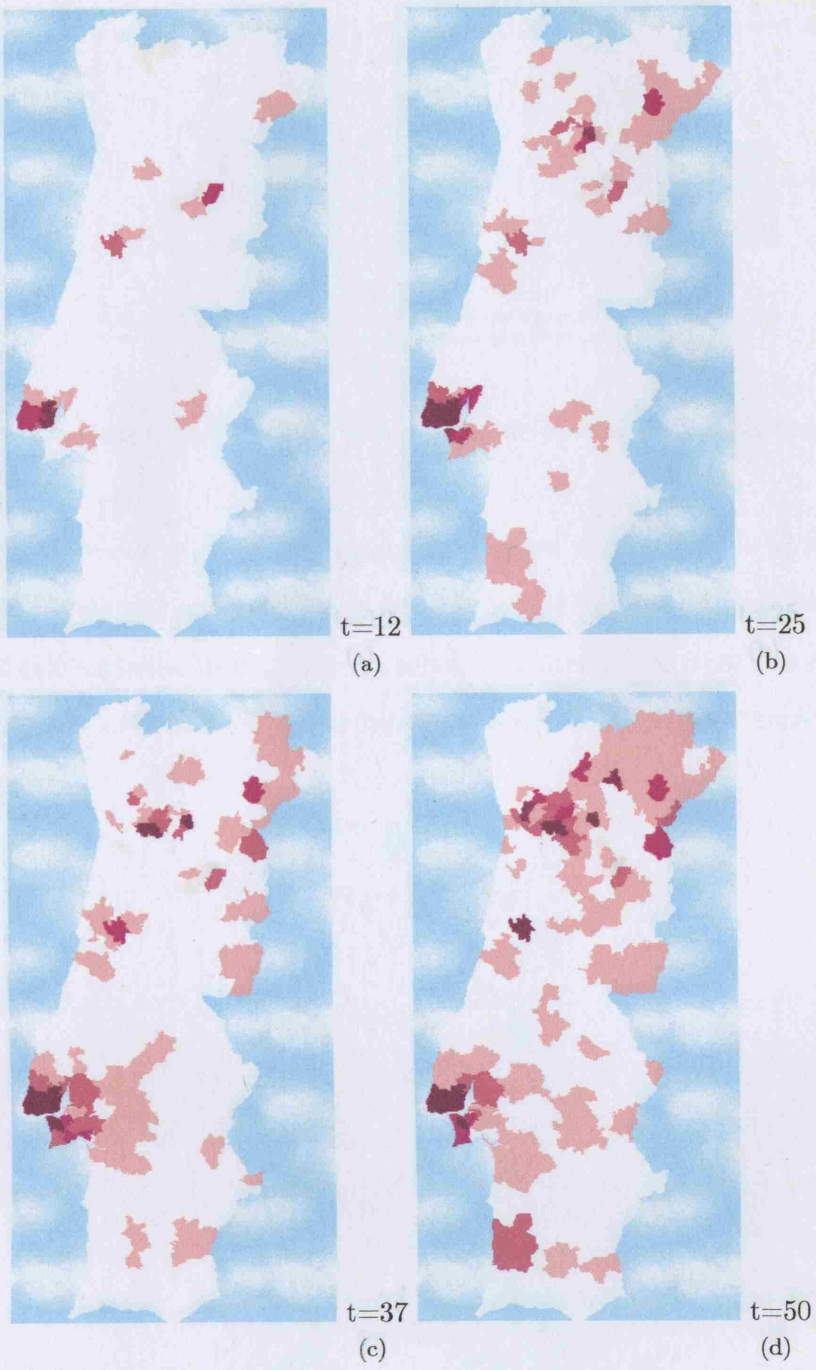


Figure 4-31: Screenshots of a simulation using infection force=0.5.

throughout the microbial population[36]. These scenarios would introduce a evolutionary feature in the model, which is not present at this moment, as we do not consider ihneritance for the evolution of β.

In figure 4-33, there is a comparison between the evolution of the epidemic size con-

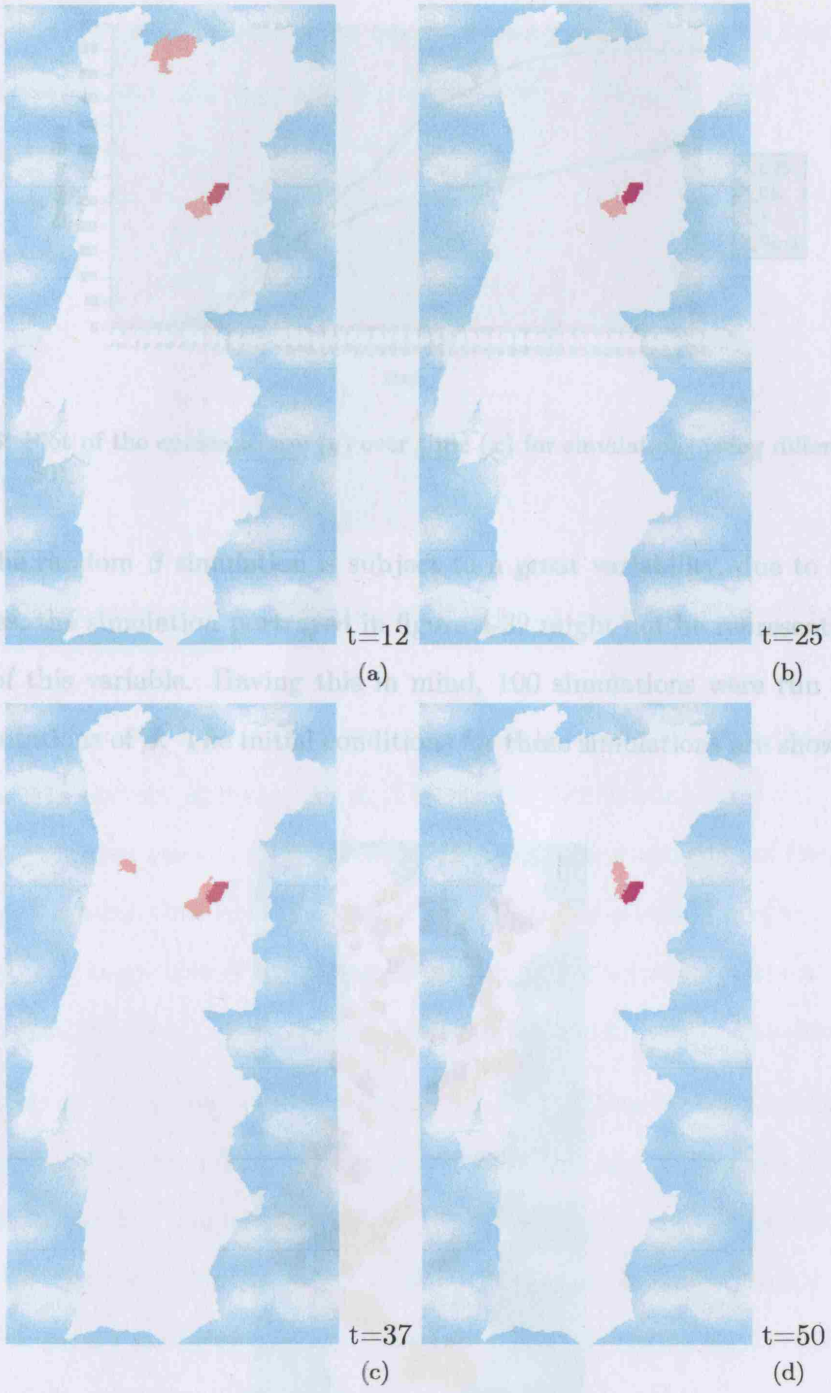


Figure 4-32: Screenshots of a simulation considering random mutations of the infection force.

sidering different rates of infection. As expected, the higher epidemic size corresponds to the deterministic β (β=1), followed closely by the 0.5 β. In the cases of β=0.25 and random β, the impact of the disease is severely reduced and there is no evident outbreak.

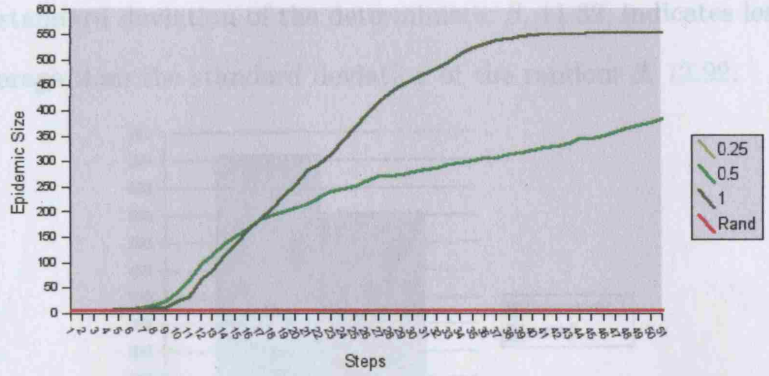


Figure 4-33: Plot of the epidemic size (y) over time (x) for simulations using different infection forces (t=0,..,50).

Since the random β simulation is subject to a great variability, due to its inherent randomness, the simulation portrayed in figure 4-32 might not be representative of the behavior of this variable. Having this in mind, 100 simulations were run considering random mutations of β. The initial conditions for these simulations are shown on figure 4-34

Figure 4-34: Initial conditions for the simulations using a random infection force (t=0).

The average epidemic size across these simulations is shown in figure 4-35 and compared to the average epidemic size with a unitary β. These results show a much smaller epidemic size for the random β, compared to a deterministic β=1. Also, as it would be

expected, the standard deviation of the deterministic β, 11.32, indicates less dispersion around the average than the standard deviation of the random β, 12.92.

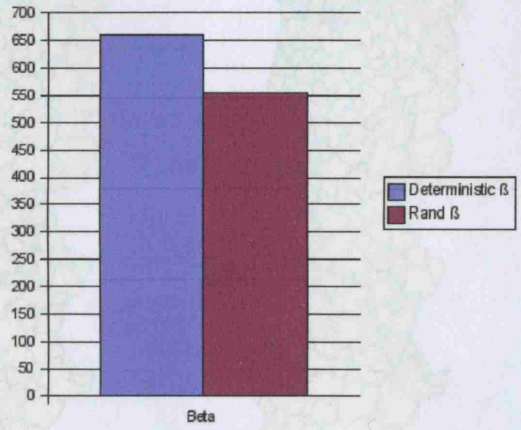


Figure 4-35: Comparison of the average epidemic size with a deterministic (β=1) and a random infection force (100 simulations).

The maps in figure 4-36 show the average distribution of affected individuals in the simulations with unitary and random β. The spatial distribution does not differ much and the same clusters emerge on both maps. However the magnitude of these values is very different. We can thus conclude that β is a important parameter of the model, but it influences the magnitude of the epidemic rather than its spatial pattern. Moreover, using a deterministic β=1, may strongly overestimate the impact of the disease.

The length of the latent period - transition from susceptible to infectious - and morbid period - transition from infectious to removed - are parameters that regulate a certain delay on the transmission of the infection. In the case of latent period, its length determines the lapse of time in which infectious individuals cannot spread the disease which means the longer it is, the longer will be the time range required for the occurrence of the epidemic. The initial distribution of infectious individuals used in the simulation is represented in figure 4-37.

One way of isolating the influence of this parameter is to consider simulations with no morbid period; *i.e.*, the phase transitions are limited only to susceptible to latent and latent to infectious. The trick to induce this behavior is to consider a morbid period longer than the simulation time. Figures 4-38, 4-39 and 4-40 show simulations

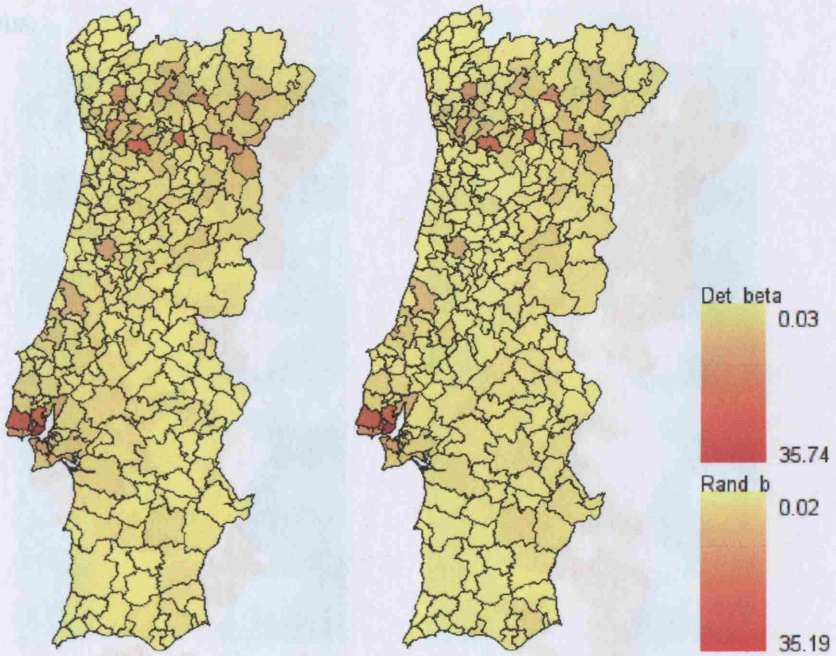


Figure 4-36: Average distribution of affected individuals in simulations: with a) infection force=1 and b) random infection force (100 simulations).

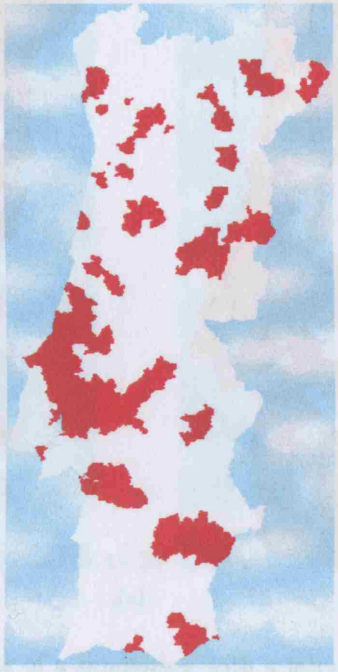


Figure 4-37: Initial conditions for the simulations concerning the latent period (t=0).

using three different latent periods: 1, 14 and 20 time steps. Although employing different time steps, all simulations tend to reproduce the same pattern of distribution

of infectious.

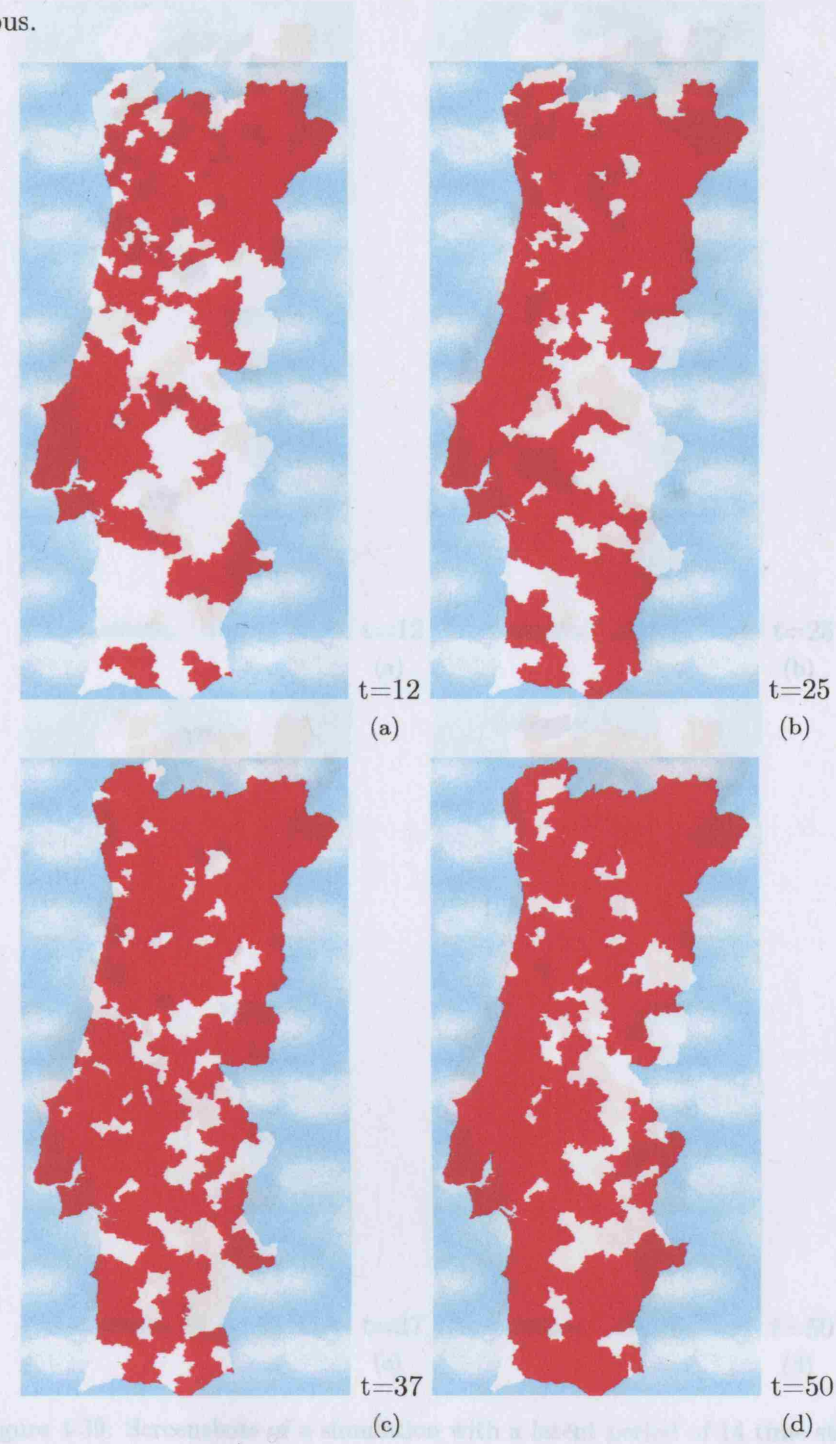


Figure 4-38: Screenshots of a simulation with a latent period of 1 time step.

The graphs of the evolution of population categories that we see in figures 4-41 help us to understand this behavior more clearly. On the three graphs, an increment in the number of infectious individuals once the latent period is over, is quite visible. That

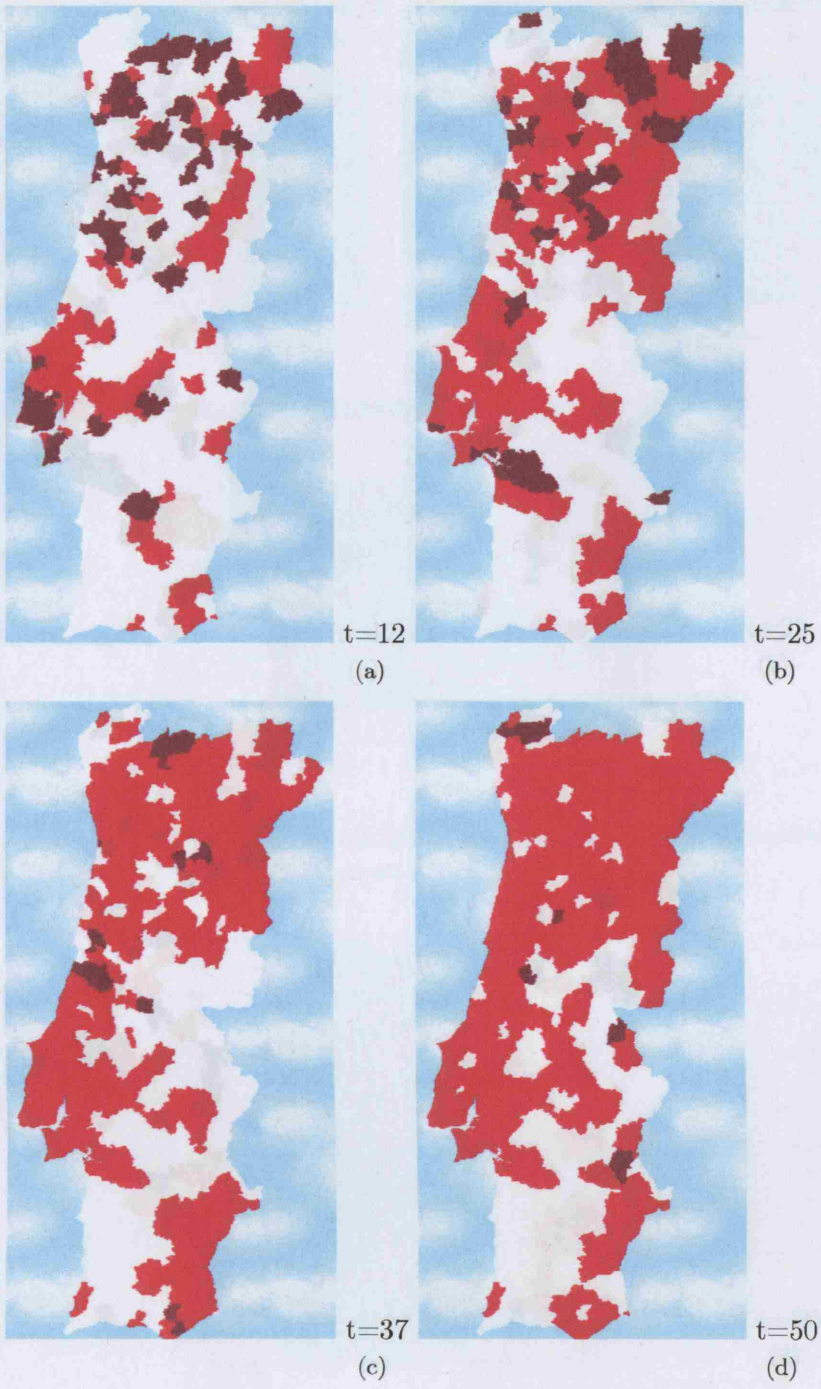


Figure 4-39: Screenshots of a simulation with a latent period of 14 time steps.

increment starts earlier in the simulation with the shortest latent period, in figure 4-41a and later in the simulation with the longest latent period in figure 4-41c. Nevertheless, all experiments tend to the same value of infectious individuals and it is only a matter of time before they reach it.

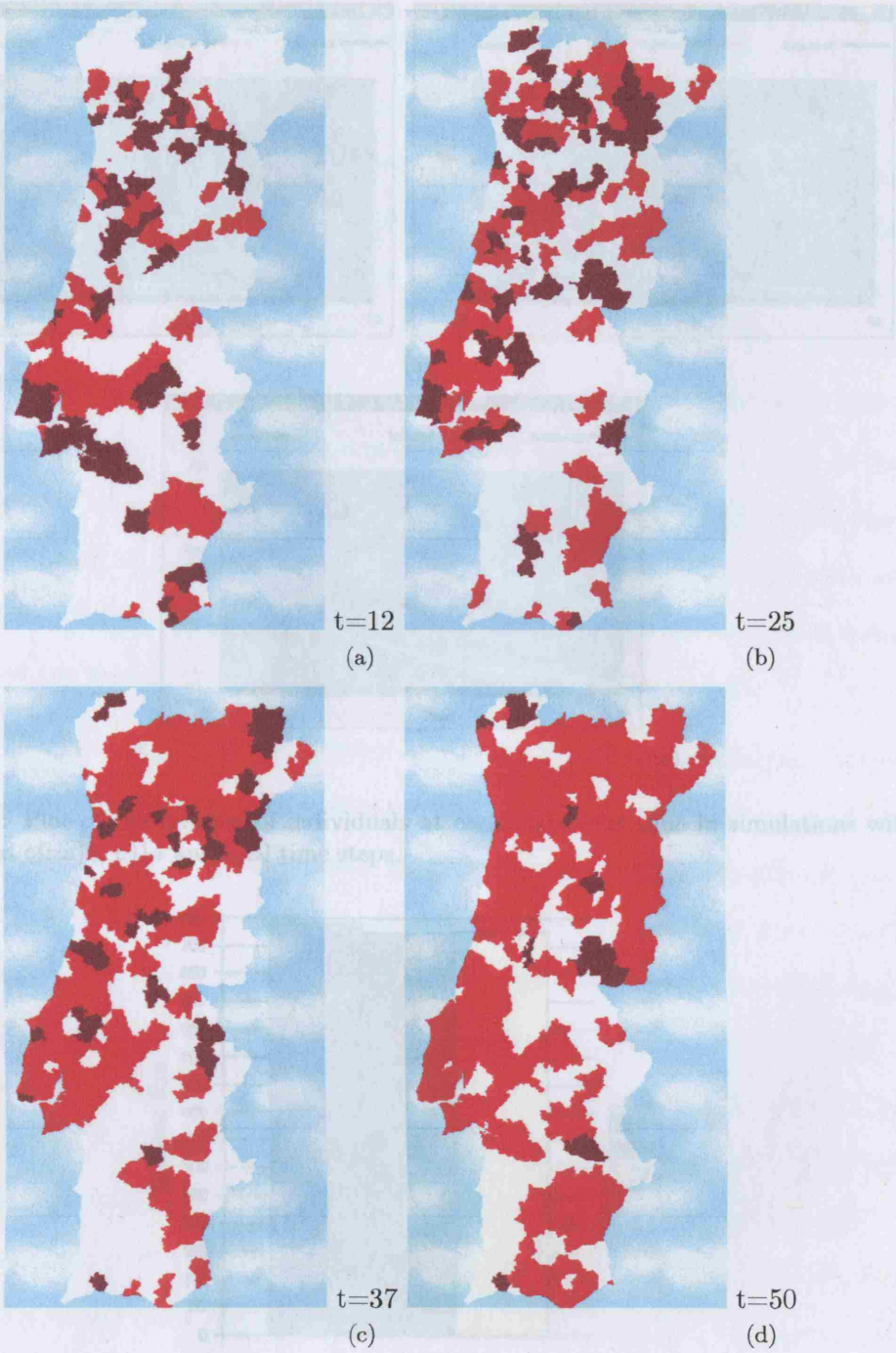


Figure 4-40: Screenshots of a simulation with a latent period of 20 time steps.

These observations are supported by the histogram on figure 4-42; in fact, the final epidemic sizes are very similar in the three simulations, despite the duration of the latent period.

From these simulations, it can be said that the latent period has the role of delaying

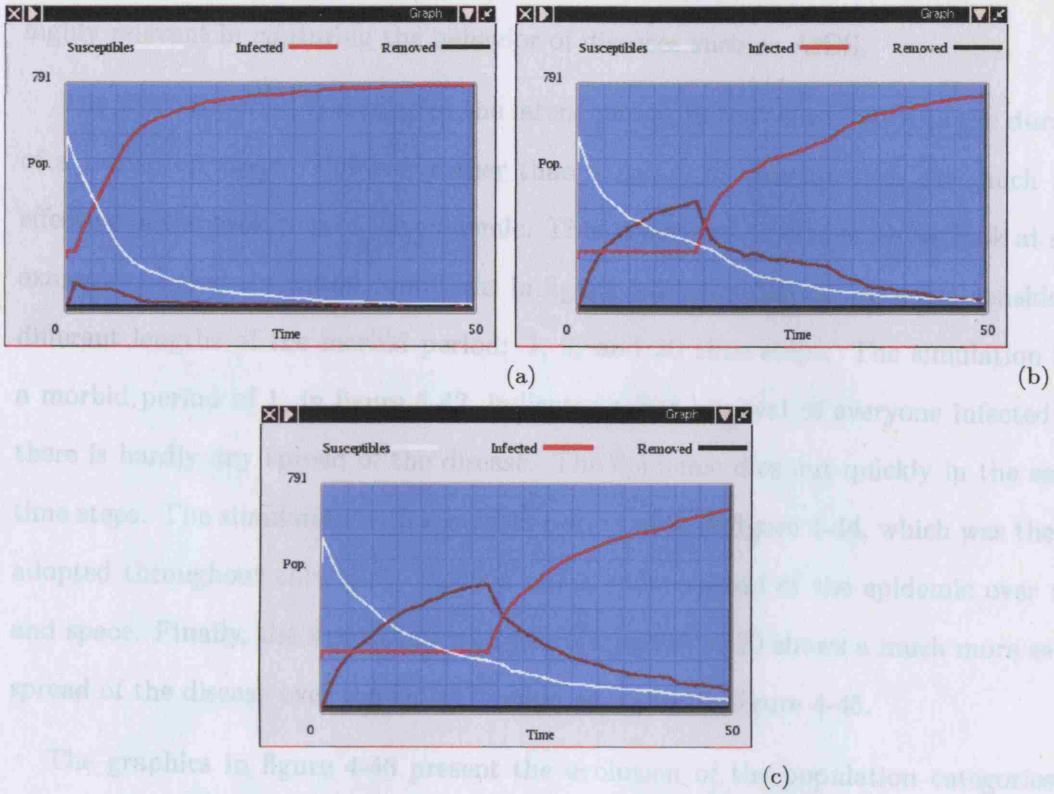


Figure 4-41: Plot of the number of individuals at each state over time in simulations with a latent period of: a)1, b)14 and c)20 time steps.

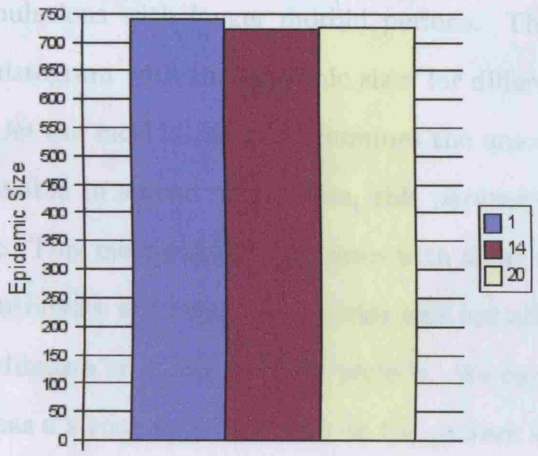


Figure 4-42: Comparison between the final epidemic size (t=50) in simulations with different latent periods.

the spread of the epidemic rather than regulating its spatial pattern or the epidemic size. However the results of this section suggest that the study of the latent period is important, especially when combined with the morbid period, and this parameter is

highly relevant in capturing the behavior of diseases such as AIDS.

The morbid period is similar to the latent period in the sense that it is the duration of a transition stage. However rather than a delay, its consequences are much more effective in the evolution of the epidemic. This is perhaps clearer when we look at some examples. Using the initial conditions in figure 4-37, simulations were run considering different lengths of the morbid period: 1, 9, and 20 time steps. The simulation with a morbid period of 1, in figure 4-43, indicates a fast removal of everyone infected and there is hardly any spread of the disease. The epidemic dies out quickly in the earlier time steps. The simulation with a morbid period of 9, in figure 4-44, which was the one adopted throughout this work, shows a much wider spread of the epidemic over time and space. Finally, the simulation with morbid period of 20 shows a much more severe spread of the disease over the entire domain as we see in figure 4-45.

The graphics in figure 4-46 present the evolution of the population categories for the previous simulations. It is possible to observe that for the shortest morbid period, the increment of removed individuals starts earlier but does not reach the magnitude that it does in simulations with longer morbid periods. This can also be confirmed by looking at the histogram with the epidemic sizes for different morbid periods as we see in figure 4-47. As the morbid period determines the amount of time during which individuals are available to spread the disease, this parameter influences directly the size of the epidemic. This means that for diseases with short morbid periods, although highly lethal, the outbreaks will cease very quickly and not allow the epidemic to reach the proportions of diseases with long morbid periods. We can therefore conclude that the morbid period has a strong influence both on the pattern and the magnitude of the epidemic and that fitter virus and bacteria are going to evolve in the direction of longer morbid periods.

This section has covered some of the mechanics of the infection model described in section 2.2.2 (page 59). In this kind of model, experiments and interpretation are the best way to understand such models rather than through a methodical analysis of its formulation. The parameter values and combinations between parameters are very large

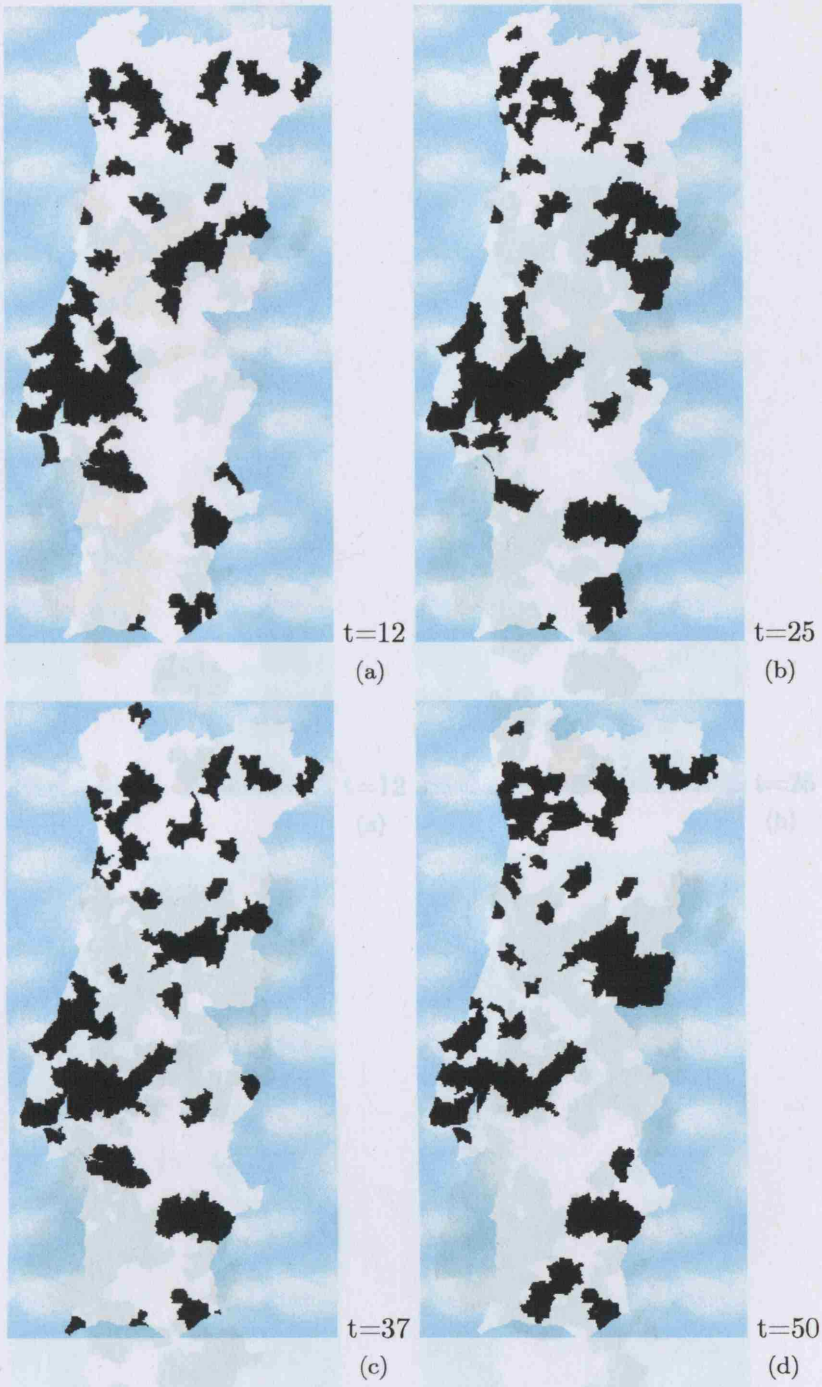


Figure 4-43: Screenshots of a simulation with a morbid period of 1 time step.

and what we have shown is only a small part of what is possible through these model experiments. However although the infection model is kept very simple in the rest of this work, we have shown how we can explore some of its complexity and how we can use flexibility to reproduce the behavior of different kinds of disease.

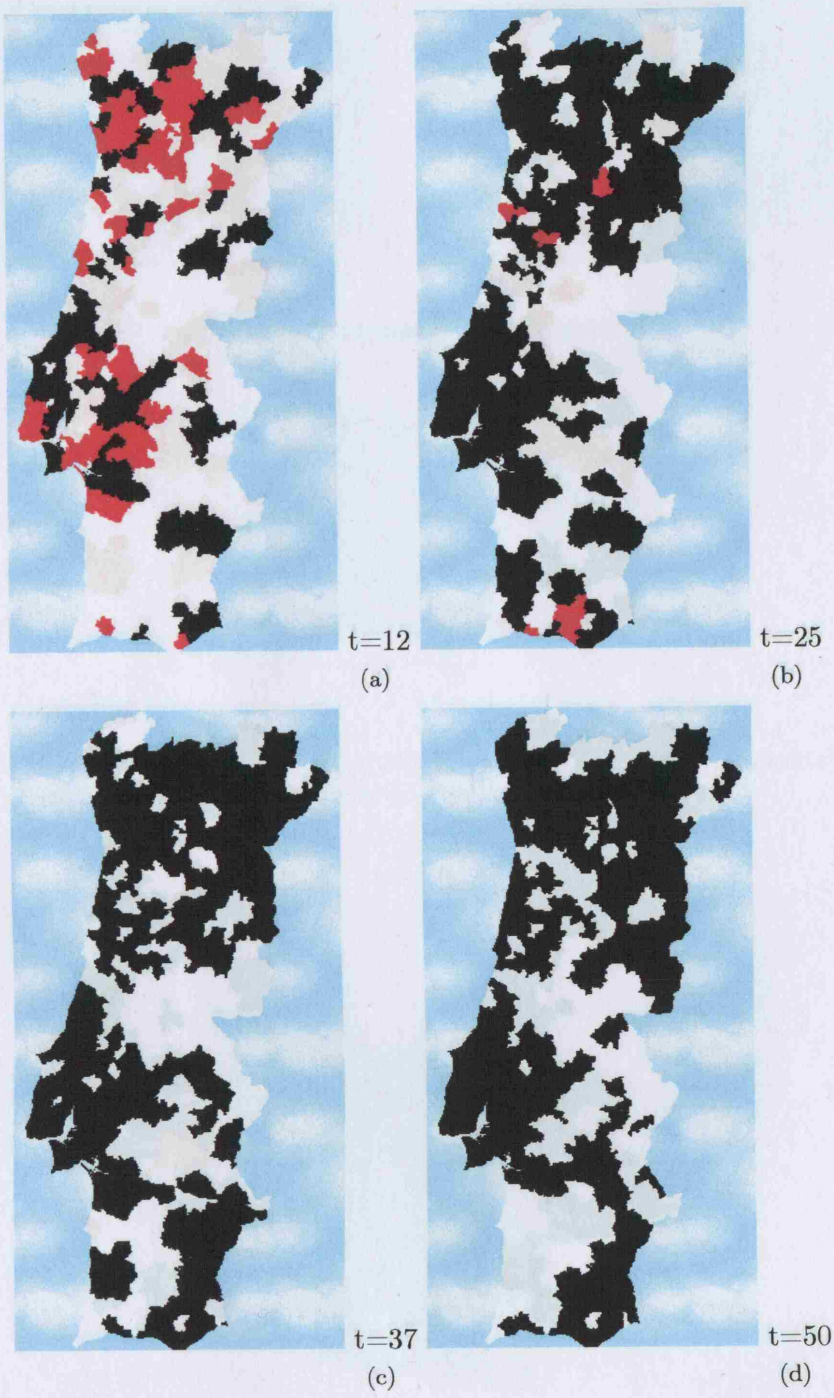


Figure 4-44: Screenshots of a simulation with a morbid period of 9 time steps.

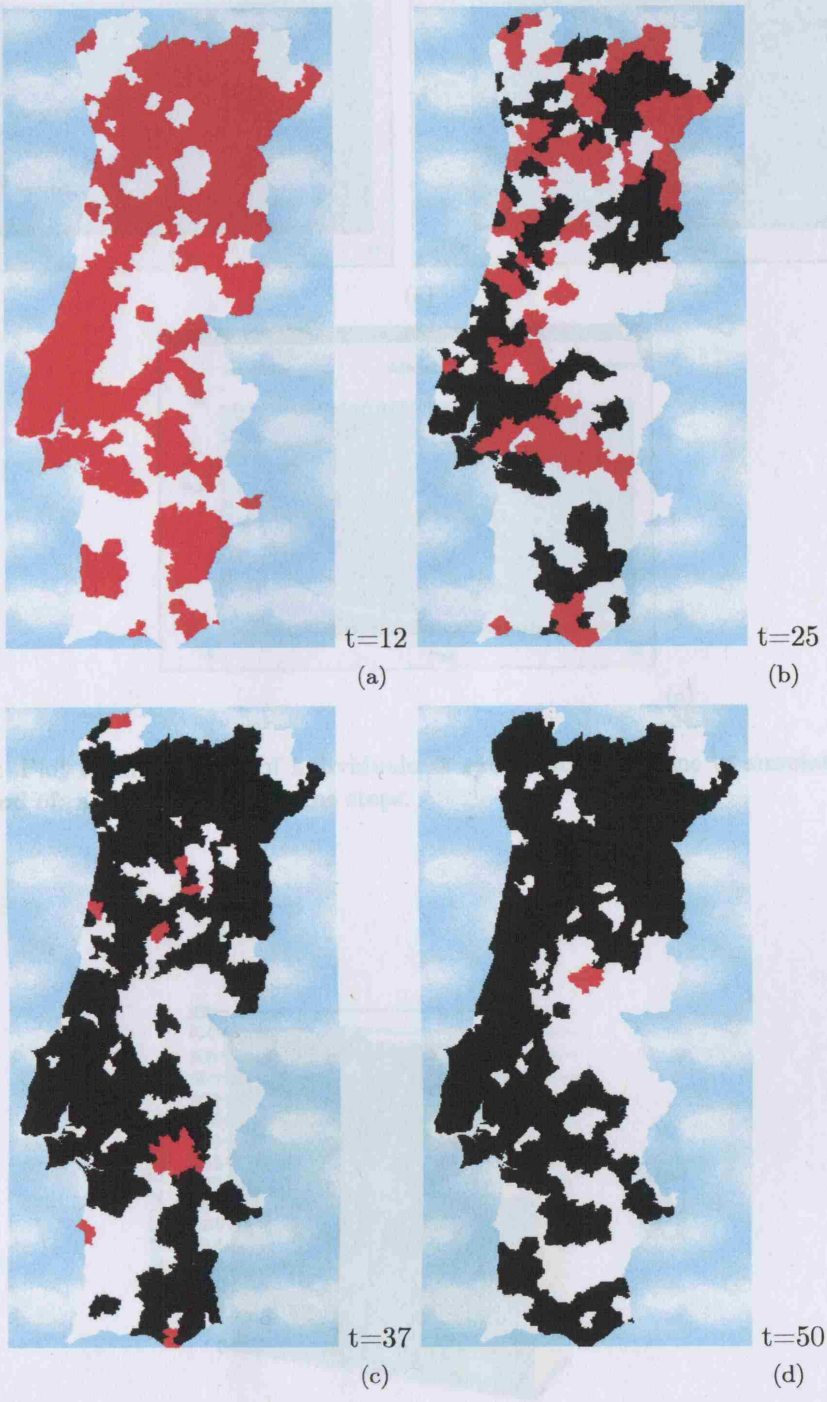


Figure 4-45: Screenshots of a simulation with a morbid period of 20 time steps.

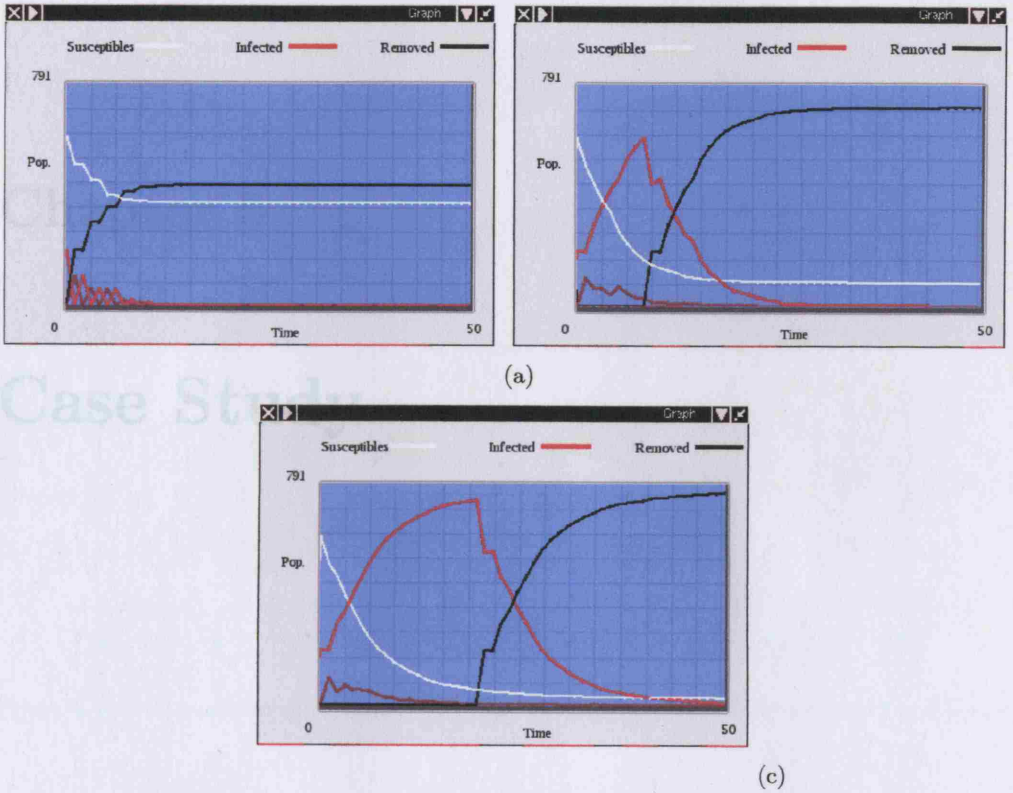


Figure 4-46: Plot of the number of individuals at each state over time in simulations with a morbid period of: a)1, b)9 and c)20 time steps.

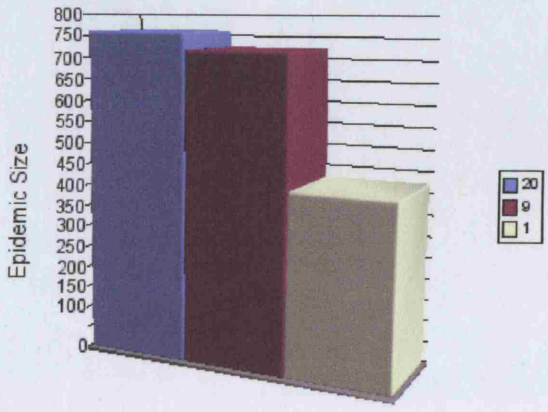


Figure 4-47: Comparison of the final epidemic size (t=50) in simulations with different morbid periods: 1,9 and 20 steps.

Chapter 5

Case Study

5.1 Presentation of the Data

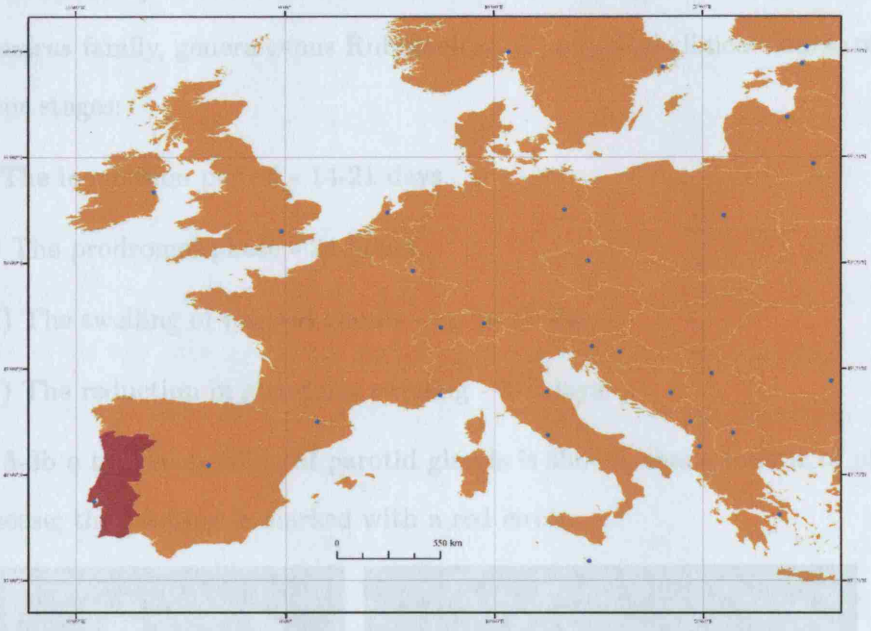
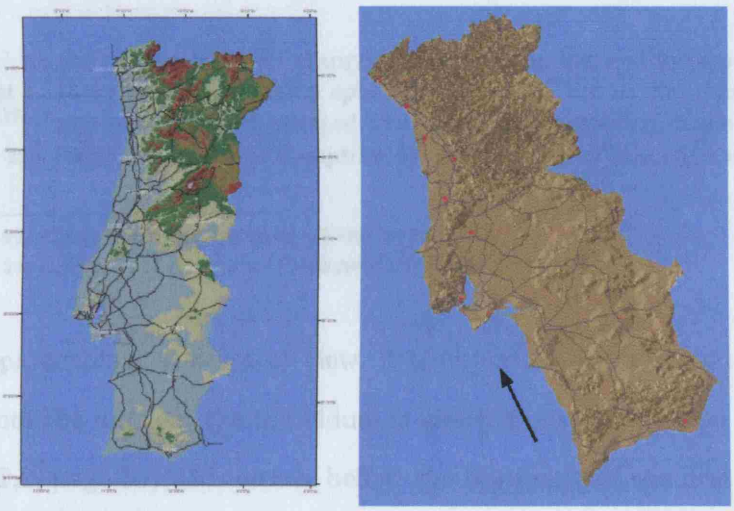


Figure 5-1: A framing map of the study area: Portugal (in red) in the context of Europe.

The case study for this model is an epidemic outbreak of mumps that took place in continental Portugal in 1996. In figure 5-1 we can see where Portugal is located in Europe and in figure 5-2 there are more detailed views of the territory.



(a) (b)

Figure 5-2: The continental Portuguese territory. In a) we see a 2D representation, with the altitudes, main rivers and roads and in b) we see a 3D view of the Digital Terrain Model (DTM).

Mumps (infectious parotitis) is an acute infectious disease which is highly contagious. It is caused by the mumps virus which we show in figure 5-3a, a member of the *Paramyxovirus* family, genera genus Rubulavirus. The typical clinical course of mumps follows four stages:

- (I) The incubation period - 14-21 days.
- (II) The prodromal phase - 24 hours.
- (III) The swelling of parotid glands - up to 10 days.
- (IV) The reduction in glandular swelling - 3-7 days.

In figure 5-3b a typical swelling of parotid glands is shown, characteristic of phase (iii) of the disease; the swelling is marked with a red circle.

(a) (b)

Figure 5-3: In a) we can see the human Paramyxovirus that can induce illnesses such as mumps or measles: most of these virus are roughly spherical (about 200nm in diameter) but they can be much larger [37]. In b) we see a child infected with mumps: the swelling of the parotid glands, marked with a red circle, is a typical symptom from phase III of the clinical course of this disease[38].

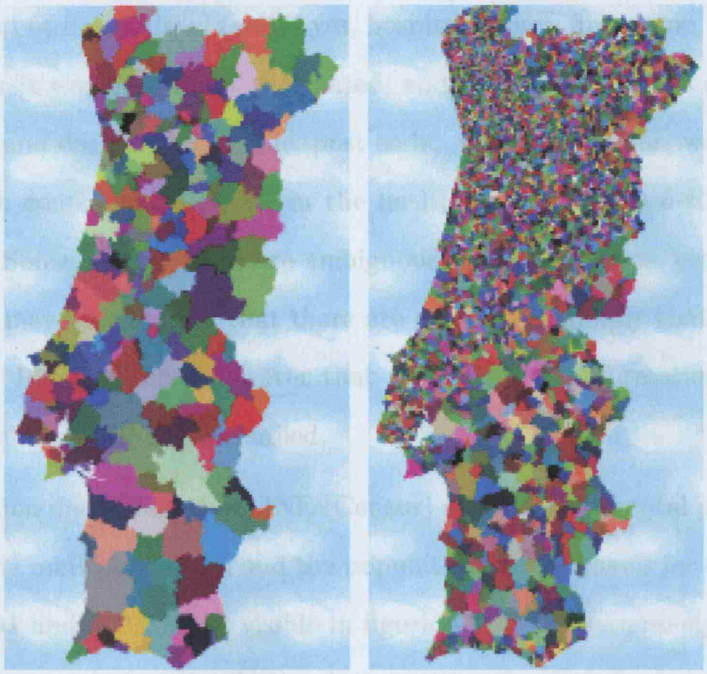
[37] `http://www.virology.net/Big_Virology/BVRNApara.html`
[38] `http://www.idph.state.il.us/about/immunepics/mumps.htm`

From the epidemiological point of view, it is important to note that mumps has a latent stage from the moment the individual is infected until the person gets infectious (see section 2.2.2, page 59); this occurs before the beginning of the first clinical symptoms. The latent period is shorter than the incubation period (10-18 days) and this is followed by an infectious period of 7-11 days. Mumps can occasionally have other consequences especially if in adults, such as orchitis, meningitis and can affect the pan-

creas or ovaries. The mumps infection is spread via respiration with the virus secreted into the mouth and throat from the salivary glands.The transmission of the mumps virus depends on close personal contact with an individual shedding the virus and it increases in overcrowded conditions. Mumps can be transmitted from one person to another from 3 days before the onset of the swelling of the parotid gland for up to a week afterwards[39]. The mumps vaccine was introduced in the 1960s and prior to this, the disease affected many children aged 5-9 years and usually before 15. It tended to be a seasonal disease (winter-spring) and epidemics usually occurred at time intervals from 2 to 5 years. By 1987, a 98% reduction in mumps incidence was reported worldwide as a direct consequence of mass vaccination.

However, an epidemic outbreak took place in Portugal in 1996-97, an episode usually attributed to vaccination failure (Dias et al., 1996; Gonçalves et al., 1998). In 1992, the Health ministry suspended acquisition of the vaccine PLUSERIX from Smith Kline due to excessive adverse reactions; PLUSERIX used the Urabe strain. Then the ministry started buying the TRIVIRATEN vaccine that uses the Rubini strain. This strain resulted in poor vaccine efficacy and outbreaks took place in heavily vaccinated populations, both in Portugal and in other countries (Dias et al., 1996; Briss et al., 1994). The data available for this study is a series of reported cases of mumps from 1989 to 2001 in continental Portugal. To estimate the susceptible population, it was necessary to compile Census data for the same time period and geographic space. Ideally this data would have the same disaggregation level as the health data although this is not the case. Considering that population changes were not very dramatic over the study period, the dataset used was however considered acceptable. Finally, it was necessary to obtain a spatial basis for the model. There are some problems with this issue, essentially related to the Modifiable Area Unit Problem (see section 3.5, page 109) and to the fact that data come in different spatial units. These have different levels of detail and thus may not have a direct correspondence between each other. According to the National Statistics Institute (INE), the official administrative division is the ancient territorial

[39] http://www.cdphe.state.co.us/dc/Epidemiology/manual/School_Guidelines.pdf



(a) (b)

Figure 5-4: The INE geographic classification of Portugal in a) *Concelhos* and b) *Freguesias*. In the maps, each geographic unit is represented with a distinct color.

classification established in Portugal and it is constituted in three hierarchical levels: *Distrito*, *Concelho* and *Freguesia* (from the largest to the smallest), each one identified by a two digit code. The administrative division code/1994 revision was approved by the deliberation n 86 of 15/12/1994 of the Concelho Superior de Estatistica which is now the base of the code used by INE. The mumps series from 1989 to 1999 is aggregated by *Concelhos* and the series from 2000-2001 is aggregated by *Freguesias* which is the finer scale. Both classifications are represented in figure 5-4, with random colors attributed to each unit.

Besides this difference in spatial aggregation, the recent data also have more information about each case and are sometimes tagged with a postcode. The data from 1989-1999 contain the following relevant information for each individual: geographical location code (regiao, *Distrito*, *Concelhos*), confirmation of the information, name, gender, age, birth date and notification date. The data from 2000-2001 have, besides this information, clinic name, clinical exams, number of related cases, death, probable origin of the infection, risk activity, description of the risk activity, information whether the

case was or was not hospitalized (and if yes, hospital name), first name, surname, information whether it was or was not vaccinated, and if yes other details such as number of inoculations and date of the vaccine, post code, *Freguesia*, doctor, workplace, health delegate, health centre, and user ID in the health centre. These fields are not filled for every case. Some of the entries are ambiguous, like for instance, geographical units without values may mean either that there are no cases or simply that these have not been reported. It was noticed however that the quality of information has improved with time as well as being more detailed.

The population data comes from INE (Census) and covers the total population (men and women), the male population and the population in age classes for all *Concelhos* in Portugal in 1991 and 2001. As is visible in figure 5-5, there were no significant spatial variations in the population distribution between these years.

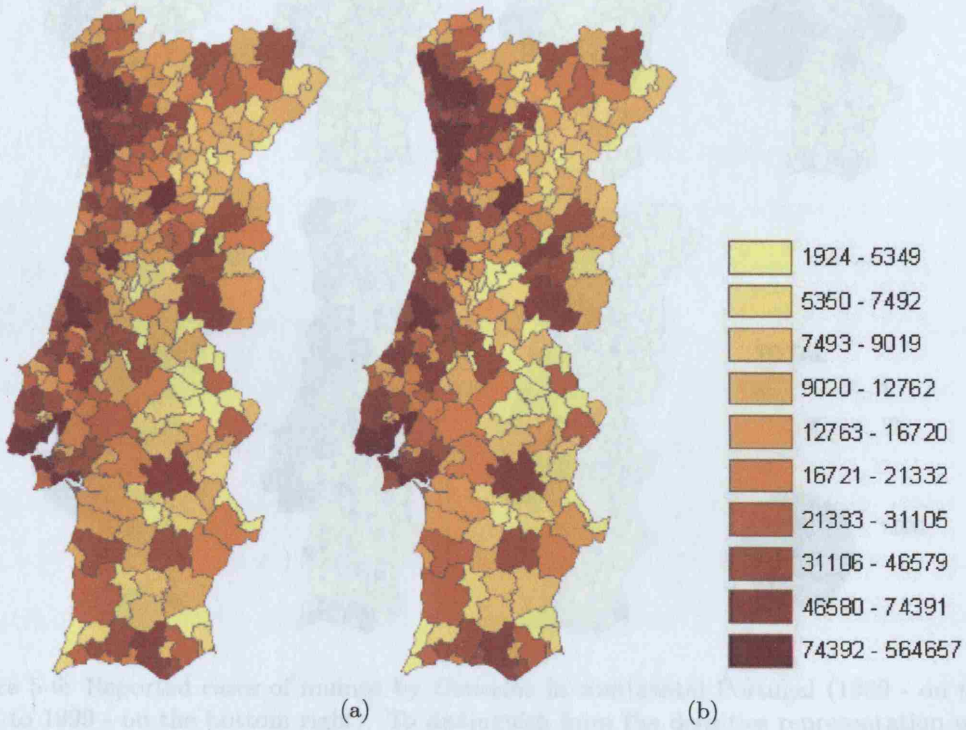


Figure 5-5: Population distribution according to the Census: in a) 1991 and b) 2001. These maps show the number of inhabitants per *Concelho*.

In figure 5-6, the complete series of reported cases of mumps from 1989 to 1999 is presented. However the data from 2000 and 2001 in figure 5-7 were not used as these are outside the epidemic outbreak that concerns this study.

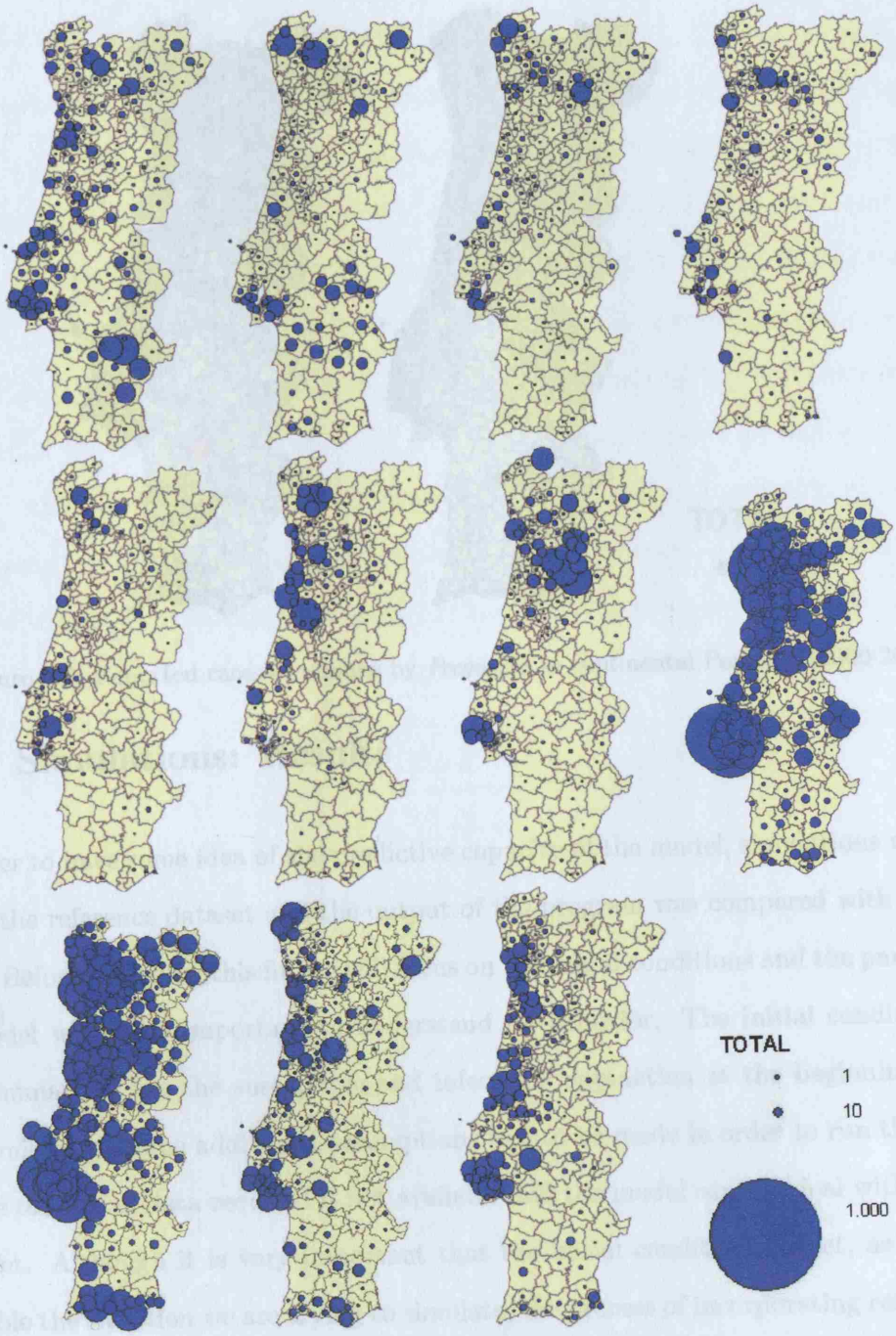


Figure 5-6: Reported cases of mumps by *Concelho* in continental Portugal (1989 - on the top left - to 1999 - on the bottom right). To distinguish from the densities representation used on the population maps (figure 5-6), the maps reporting infectious cases adopt a proportional circle representation which uses a radius proportional to the number of infected people.

The preparation and assembling of these datasets in a database and its integration in the application, has already been explained in section 3.5 (page 109).

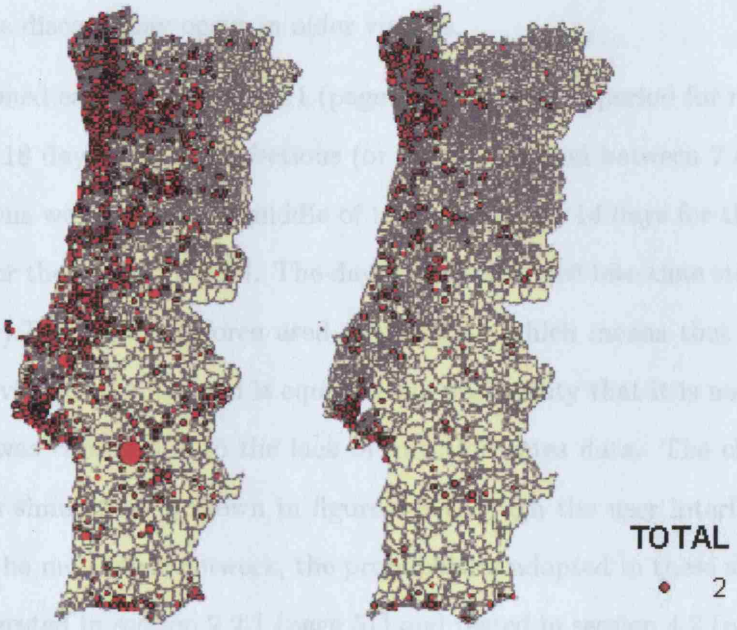


Figure 5-7: Reported cases of mumps by *Freguesia* in continental Portugal (2000-2001).

5.2 Simulations: Results

In order to have some idea of the predictive capacity of the model, simulations were run using the reference dataset and the output of the program was compared with the real data. Before analyzing this fit, we will focus on the initial conditions and the parameters of model which are important to understand its behavior. The initial conditions for the simulations are the susceptible and infectious population at the beginning of the epidemic case. Some additional assumptions had to be made in order to run the model as the totality of data required is not available and the model can not deal with the full dataset. Although it is very important that the initial conditions reflect, as much as possible the situation we are trying to simulate, the process of incorporating real data in a model is a difficult task, that often demands a lot of simplifications. For the infectious population, the reported cases of mumps in 1993 was used, which may be only an under estimate of the real values for not all cases are reported. For the susceptible population, unfortunately there is no data for this year so we used the population from the previous census (1991) which is close in time. As mumps is mainly a school-age child disease, the susceptibles are the population aged under 14 years which may also not be entirely

correct as the disease may occur in older victims.

As mentioned earlier in section 5.1 (page 161), the latent period for mumps can vary from 10 and 18 days, and the infectious (or morbid) period between 7 and 11 days. In the simulations we adopted the middle of these intervals: 14 days for the latent period and 9 days for the morbid period. The days were converted into time steps (1 time step equals 1 day).The infection force used (β) was 0.5 which means that the probability that the individual gets infected is equal to the probability that it is not infected. This assumption was chosen due to the lack of infection rates data. The choice of parameters for this simulation is shown in figure 5-8 through the user interface of EPISIM. Concerning the movement network, the probabilities adopted in these simulations were the ones suggested in section 2.2.1 (page 51) and tested in section 4.2 (page 117), as the combination of the four parameters using a distance decay law.

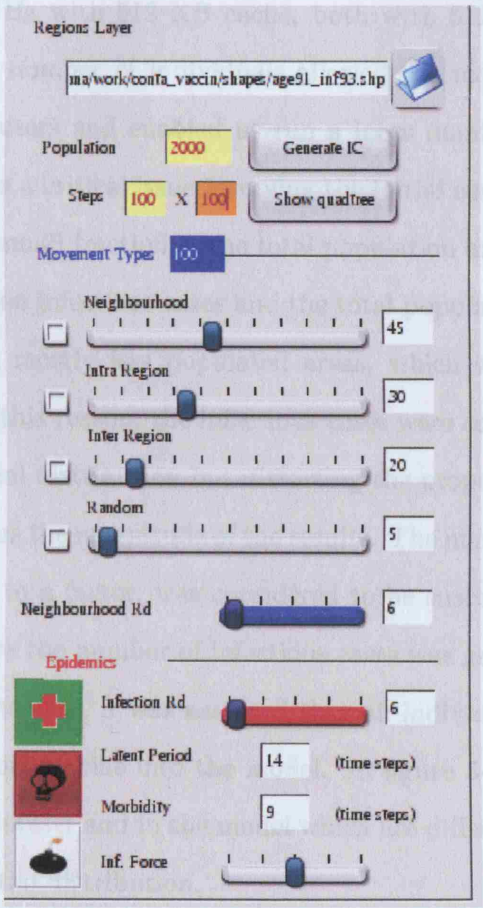
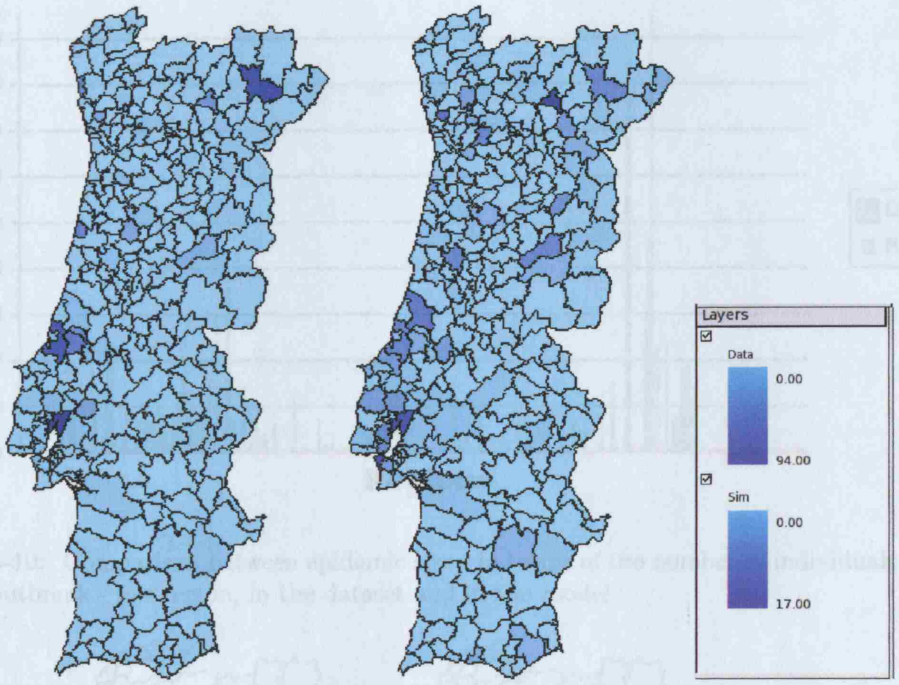


Figure 5-8: Choice of parameters for the simulations.

The total calculated susceptible population of Portugal is 1584 000 habitants, which is a extremely high value. Although the model is programmed in C++ outside any GIS platform, and optimization techniques were used to improve its efficiency (see section 3.4, page 92), the fact that this is an agent-based model, means that decisions and rules are focused on each individual which makes computation very intensive. For this reason only 792 individuals were simulated, which means there was an aggregation based on the fact that each individual in the model represents 2000 individuals in reality. We should note that this limitation is related to the hardware that was used and it is not a limitation of the model or of the program itself. In fact as it was stated in section 3.1 (page 76), the model was programmed using *threads* which would enable it to take advantage of a dual processor. Unfortunately, the computers used for the simulations were not very powerful - a Intel Celeron 1300 MHz with low cache (256 KB) and a PowerPC G4 1.33 GHz with 512 KB cache, both with 512 MB RAM memory. This simplification in the number of individuals allowed the model to run at considerable speed in these computers and enabled us run a large number of simulations. On the other hand and this is a critical issue involving the initial conditions, the reported cases of mumps are a very small fraction of the total population and scaling them to preserve the proportion between infectious cases and the total population, would eliminate most of them; this affects mostly less populated areas, which would not be assigned any infectious cases. For this reason, the infectious cases were considered in absolute value, preserving their spatial distribution but distorting the proportion between susceptibles and infectious and thus the magnitude of the results. The non infectious population, this one scaled according to a factor, was considered to be susceptible. This simplification led to situations where the number of infectious cases was greater than the population, and to avoid inconsistencies, it was assumed that all individuals were infectious which introduced another sort of bias into the model. In figure 5-9, we represent the initial infected cases in the dataset and in the model which are different in magnitude but have roughly the same spatial distribution.

One hundred simulations were run with the conditions described in the previous



(a) (b)

Figure 5-9: Initial Conditions: a) in the dataset (1993) and b) in the simulations (t=0). These maps refer to the initial distribution of infectious individuals.

paragraphs. The correlation coefficient between the epidemic size in each simulation and the observed dataset varies between 0.88 and 0.94, the average correlation being: 0.92. Equation (5.2) shows the formula used for the *Pearson Correlation Coefficient*, where z_x is the observed dataset, z_y the estimated dataset, and N the number of regions.

$$r = \frac{\sum z_x z_y}{N-1}, \tag{5.1}$$

The correlation between the average results for 100 simulations and the observed data is 0.96. Figure 5-10 shows the distribution of the epidemic size in the dataset and the average output of the model. The scales are very different and there is an under estimation of the higher values which is probably related to the biased initial conditions.

Figure 5-11 shows the final distribution of affected individuals in the dataset and in the model. Although the magnitude of the values is very different, we can identify in both scenarios, the same clusters of affected individuals.

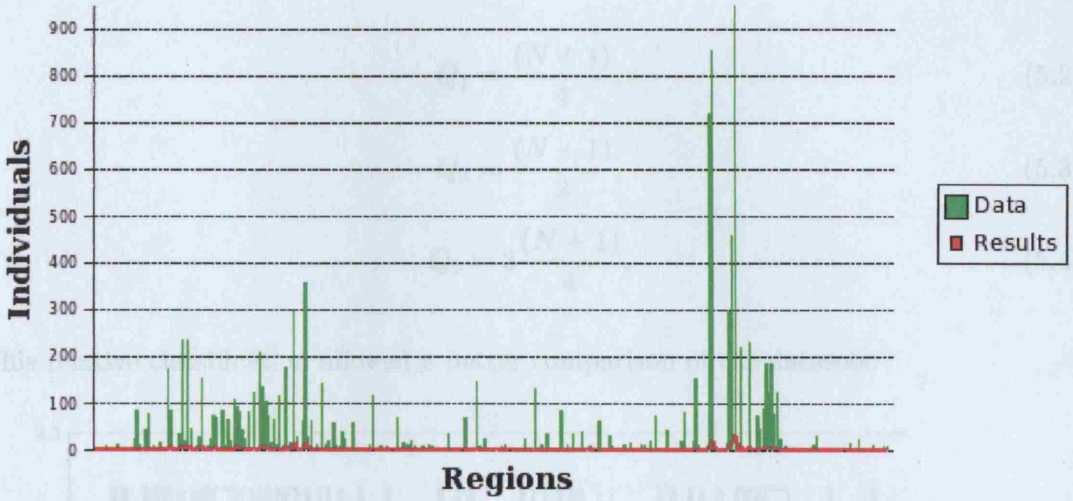


Figure 5-10: Comparison between epidemic size - in terms of the number of individuals affected by the outbreak - per region, in the dataset and in the model.

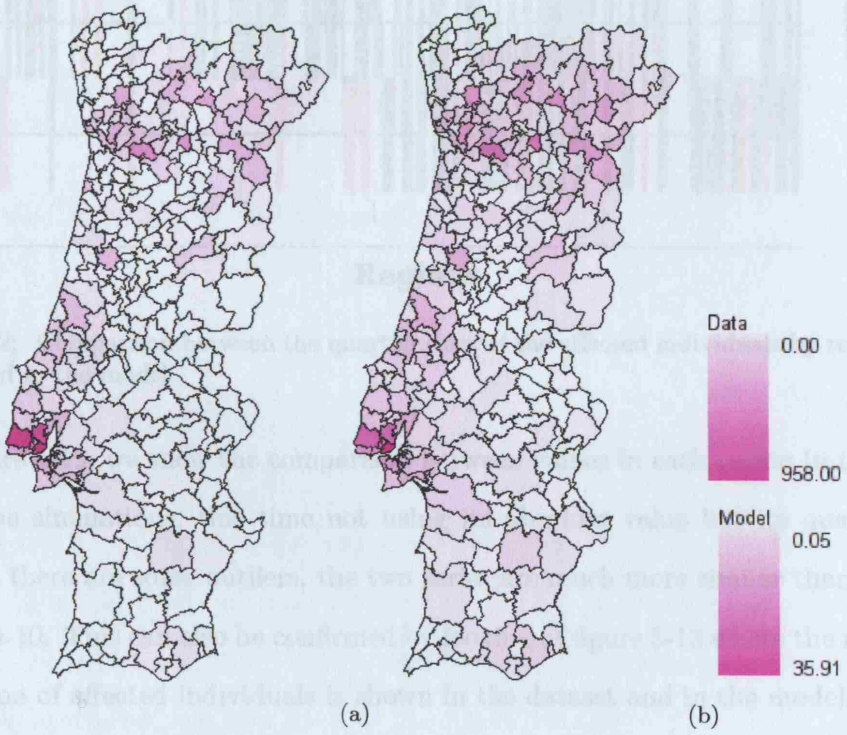


Figure 5-11: Distribution of the affected individuals at the end of the outbreak, a) in the dataset and b) in the simulations.

To remove the magnitude difference, the two series were standardized. This standardization was achieved with a classification based on the quartiles. The lower and upper quartiles were evaluated according to equations (5.2) and (5.4) and the second quartile, also known as the median, used as in (5.3).

$$Q_1 = \frac{(N+1)}{4}, \tag{5.2}$$

$$Q_2 = \frac{(N+1)}{2}, \tag{5.3}$$

$$Q_3 = 3\frac{(N+1)}{4}, \tag{5.4}$$

This relative classification allowed a better comparison of the datasets.

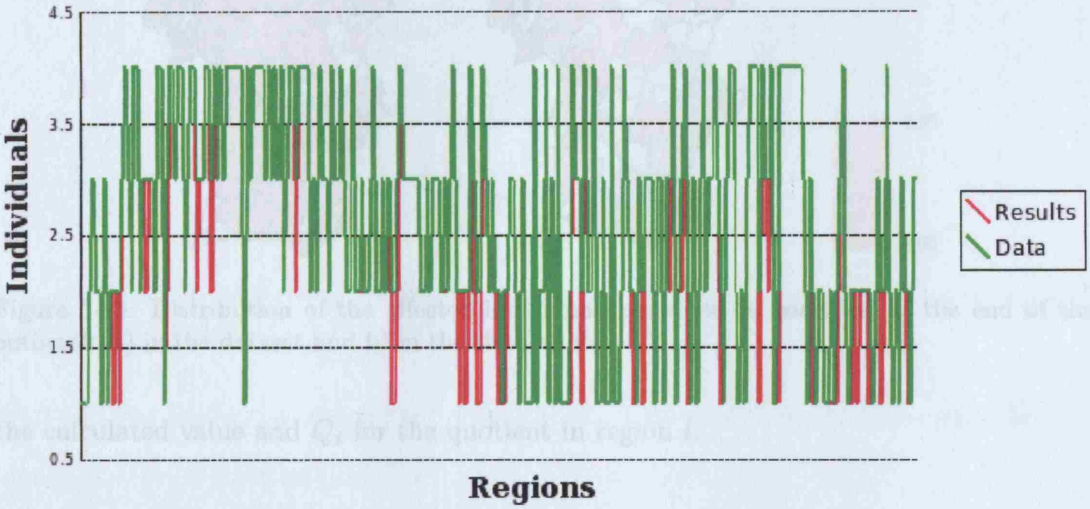


Figure 5-12: Comparison between the quartile class of the affected individuals by region, in the dataset and in the model.

In figure 5-12, we show the comparison between values in each region in the dataset and in the simulations, this time not using its absolute value but its quartile class. Although there are some outliers, the two series are much more similar than the series in figure 5-10. This can also be confirmed by looking at figure 5-13 where the normalized distribution of affected individuals is shown in the dataset and in the model.

To have a better perception of which regions produce better results, a map with the quotient between the value in the dataset and the value in the simulation for each region, was produced. To allow a relative value smaller than 1 and since the order is irrelevant, the smaller value is always divided by the greater; in this way, the closer the quotient is to 1, the closer the two values are, being 1 if they are equal. This is described in equations (5.5) and (5.6) where O_i stands for the observed value, C_i for

Figure 5-13: Distribution of the affected individuals classified in quartiles, at the end of the outbreak, a) in the dataset and b) in the simulations.

the calculated value and Q_i for the quotient in region i.

$$O_i > C_i \Longrightarrow Q_i = \frac{C_i}{O_i}, \tag{5.5}$$

$$C_i > O_i \Longrightarrow Q_i = \frac{O_i}{C_i}, \tag{5.6}$$

Figure 5-14 shows that in many cases, the values are 1 and the quotient is never less than 0.5. From this, we can conclude that the model produces good estimates of the spatial distribution of affected individuals although the magnitude of the epidemic size is biased by the initial conditions.

Finally, some remarks need to be made about the nature of these simulations. A model dealing with many random components usually faces a great variability in its results. The results shown in this section rely on average values of a large number of simulations run over time; however, to guarantee its validity, the distribution of frequencies has to be analyzed in order to assure it is a normal distribution and is converging to an average. Figure 5-15 shows the distribution of frequencies for the

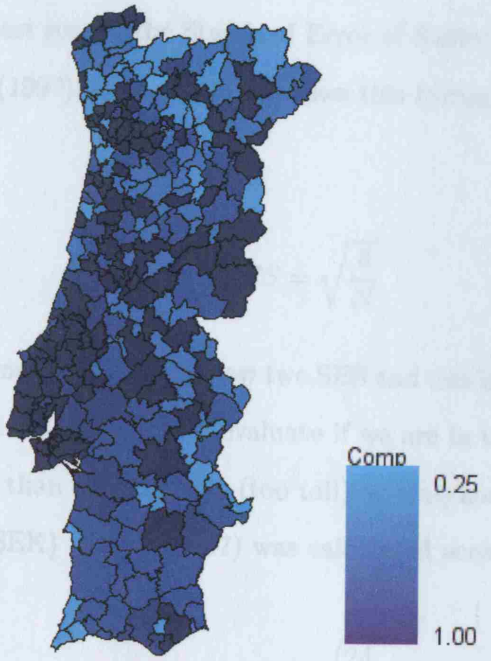


Figure 5-14: Quotient between observed and simulated values per region. This quotient varies between 0 and 1 although in this map the minimum value is 0.25.

epidemic size along the universe of simulations. This graphic is far from resembling a standard normal distribution.

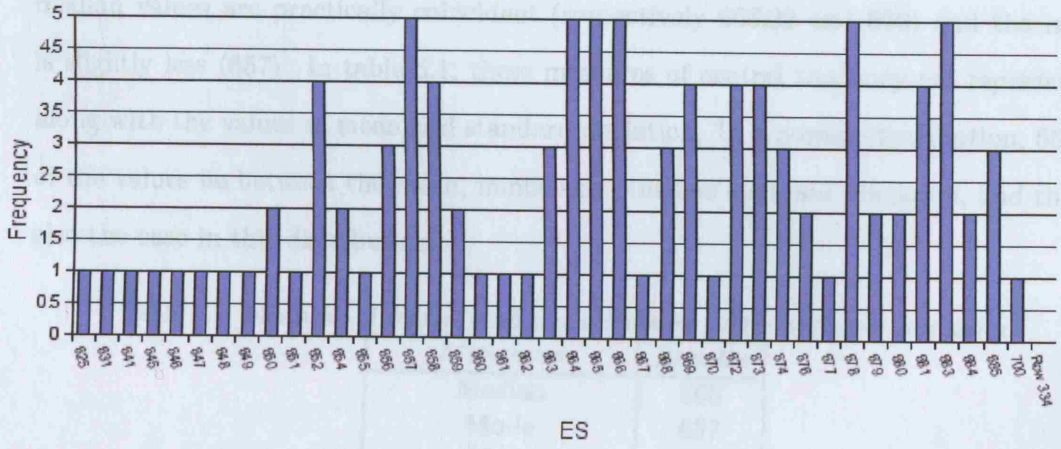


Figure 5-15: Distribution of the frequencies of the epidemic size (ES) in the simulations.

The skew coefficient of -0.37 indicates a distribution with an asymmetric tail extending towards more negative values (relative to the mean). On the other hand, the kurtosis coefficient of 0.49 indicates a relatively peaked distribution when compared to a normal distribution. To evaluate if this skewness value is acceptable for a normally

distributed set of test scores, the Standard Error of Skewness (SES) was calculated according to Brown (1997); equation (5.7) shows this formula where N is the number of observations.

$$SES = \sqrt{\frac{6}{N}} \tag{5.7}$$

An acceptable skewness should fall below two SES and this is the case, as $2xSES = 0.49$. For testing the kurtosis value and to evaluate if we are in the presence of a mesokurtic distribution, rather than a leptokurtic (too tall) or platykurtic (too flat), the Standard Error of Kurtosis (SEK) (Brown, 1997) was calculated according to equation (5.8).

$$SEK = \sqrt{\frac{24}{N}} \tag{5.8}$$

The value of kurtosis also falls bellow two SEK, $2xSEK = 0.98$, and we can say thus that this distribution is to an extent mesokurtic. Another property of the normal distribution, is the overlap of the mean, median and mode. In this case, the mean and median values are practically coincident (respectively 666.22 and 666) and the mode is slightly less (657). In table 5.1, these measures of central tendency are represented along with the values of mean and standard deviation. In a normal distribution, 66.7% of the values lie between the mean, minus and plus one standard deviation, and this is also the case in this distribution.

Table 5.1: Measures of central tendency calculated for the epidemic size series.

Measures	*Results*
Median	666
Mode	657
Mean	666.22
Standard Deviation	34.48

In conclusion, although the values generated by the model do not follow a standard normal distribution, its levels of skewness and kurtosis are fairly acceptable, and they present a normal behavior of the measures of central tendency, as well as a low dispersion around the mean. These results validate the use of average values to represent the series.

Chapter 6

Vaccination Strategies

6.1 The Optimal Control of an Epidemic

One of the most useful abilities of spatial simulation models such this one is to test strategies for the optimal control of an infection in a complex spatial landscape. One common assumption in earlier vaccination models is the homogeneity of the population; in fact, this assumption is not valid as most human populations are heterogeneous in space and in their structure, and ignorance of this fact can lead to incorrect predictions of the impact of vaccination (Keeling et al., 2003). Apart from vaccination, there are other spatial strategies for eradicating an epidemic such as quarantine. Quarantine is a control measure that can be seen as a movement restriction in the infected area or targeted at the infected individuals. Vaccination is different from quarantine for quarantine isolates the infected individuals preventing them from infecting others, while vaccination does not constitute a physical barrier to the spread of the disease. Within vaccination, the strategies can be classified as prophylactic or reactive. Prophylactic vaccination is preventive and it starts before the outbreak of the epidemic, whereas reactive vaccination assumes the knowledge of infected cases and is a directed effort. This study is focused on reactive vaccination and three strategies are simulated and compared to a normal run of the model, without vaccination. These strategies are called mass, neighborhood, and target vaccination.

6.2 Presentation of the Strategies

In figure 6-1, the initial conditions of these simulations are presented. The vaccination is introduced immediately after the outbreak of the disease where we know the location of a small group of infected individuals. It is introduced by vaccinating individuals inside the buffers (represented in blue, in figure 6-1) and the location of these buffers determines the spatial pattern of the vaccination. In technical terms, a vaccinated individual is immune which means it cannot acquire or spread the disease and thus it acts as if it is part of the removed category. During the simulation, the vaccination is not reinforced and so over time, it will only reflect the effects of this directed measure.

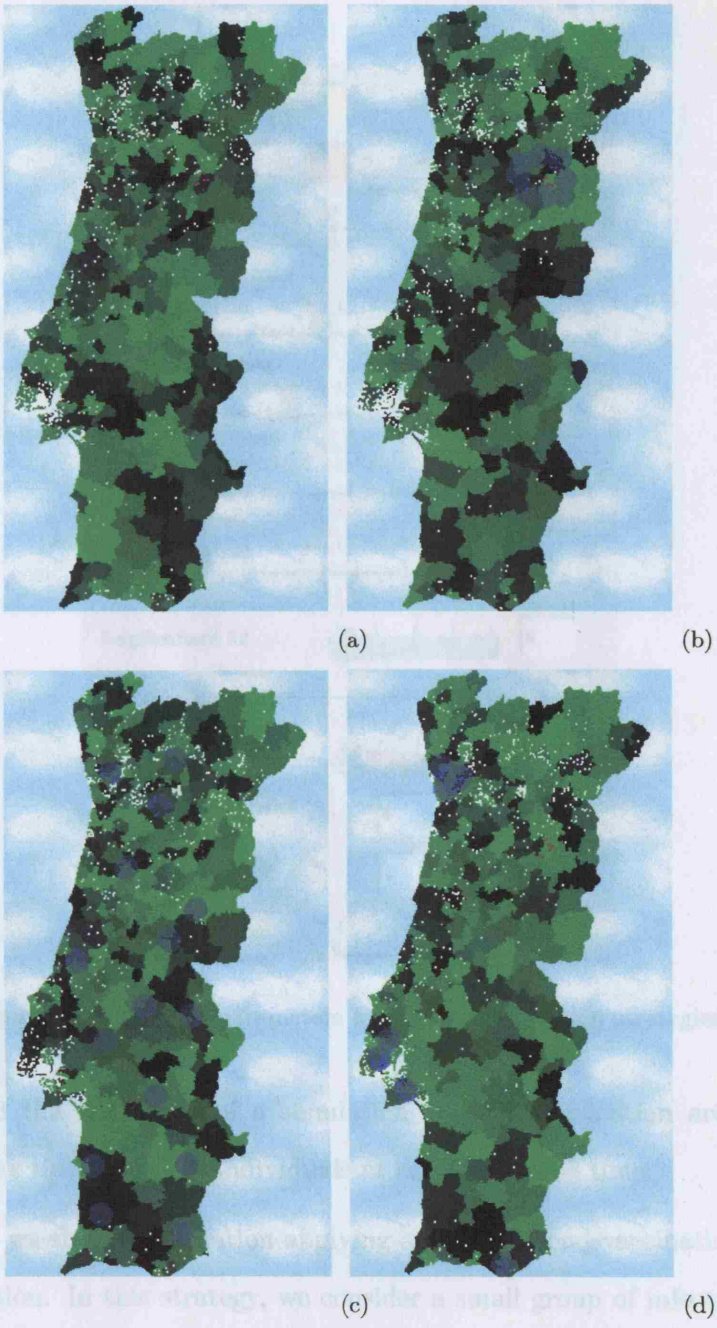


Figure 6-1: Screenshots of the initial conditions (t=0) for simulations: a) with no vaccination b) ring vaccination c) mass vaccination d) target vaccination. The vaccination buffers, areas inside which all individuals are vaccinated, are represented in blue.

Figure 6-2 shows the choice of parameters for these simulations. Because the initial number of infected individuals is very low, and to prevent the disappearance of the epidemic before it reaches an outbreak, the infectious radius was slightly increased to 8

units[40].

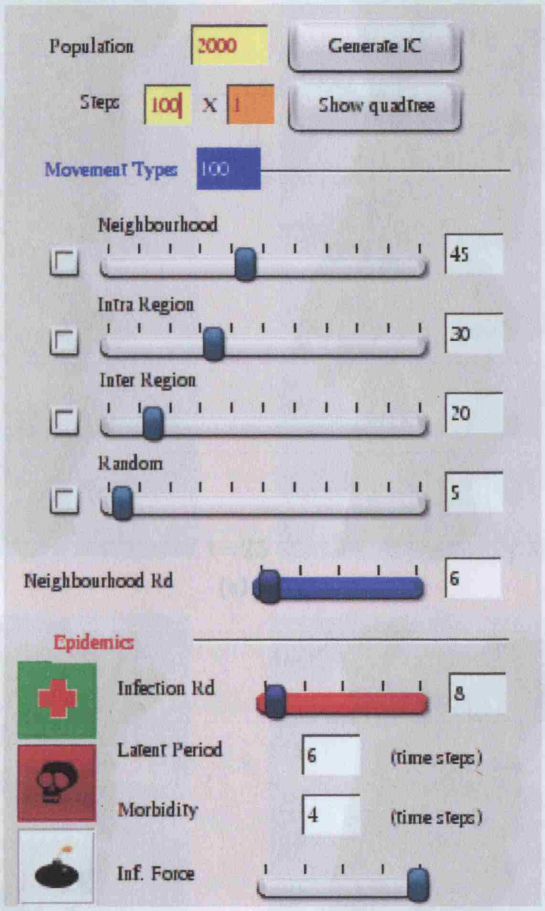


Figure 6-2: Choice of parameters for testing vaccination strategies.

In figure 6-3 the four steps of a simulation without vaccination are represented. Figure 6-4 tracks the number of individuals at each state over time.

In figure 6-5 we show a simulation applying a neighborhood vaccination also known as ring vaccination. In this strategy, we consider a small group of infected individuals inside a region, and the vaccination follows a ring around these individuals. This ring has little effect on the reduction of the epidemic for it only affects the neighborhood movement and the percolation of the infection to unaffected areas is allowed through other kinds of movement (inter-region, random). As can be observed in figure 6-3b, the ring has already dissolved by t=25 shown in figure 6-5 once the movement model starts mixing the individuals.

[40]The meaning of these units was already discussed on section 4.3 page 137.

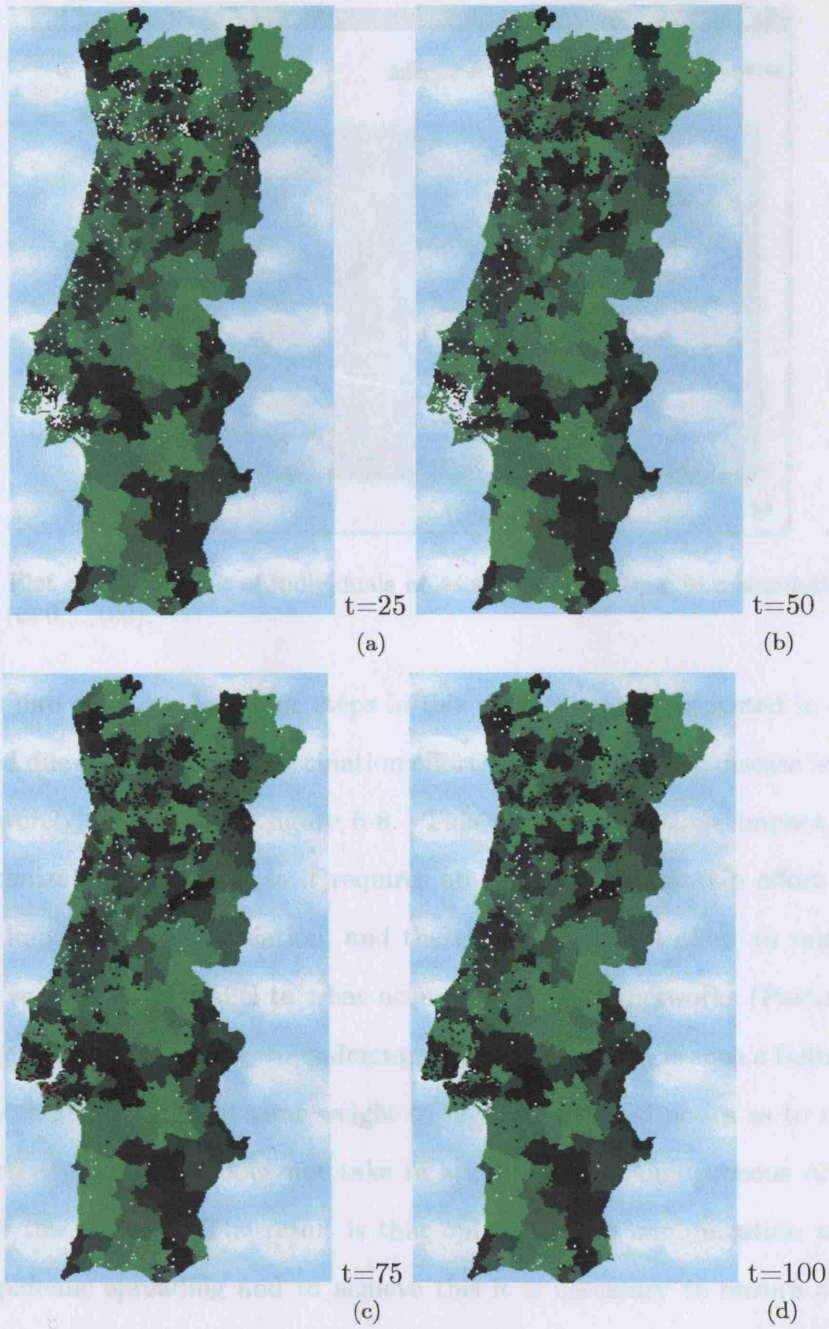


Figure 6-3: Screenshots of a simulation with no vaccination.

The results in figure 6-6 do not show any significant reduction in the number of affected individuals when compared to the simulation without vaccination in figure 6-4.

The simplest vaccination procedure would be to introduce random vaccination in order to get a uniform immunization density (Pastor-Satorras and Vespignani, 2002). This strategy is called mass vaccination for it introduces a large amount of immune

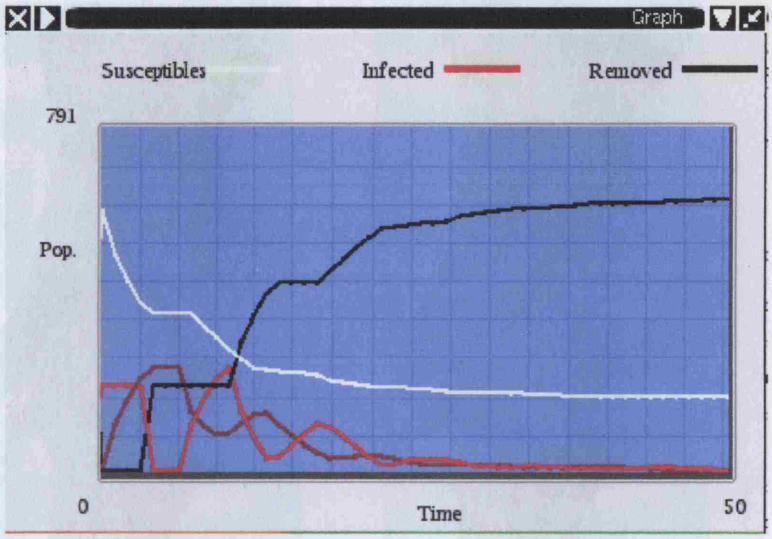


Figure 6-4: Plot of the number of individuals at each state over time in a simulation with no vaccination (t=0,...,100).

individuals into the domain. Some steps in this simulation are presented in figure 6-7. As expected due to the massive vaccination effort, the impact of the disease is restricted but not severely, as we see in figure 6-8. This strategy has some impact, but it is not an optimization technique as it requires an intensive vaccination effort to have a significant impact in the population, and therefore it does not allow to minimize the number of vaccines. In parallel to what occurs in scale free networks (Pastor-Satorras and Vespignani, 2002), it is easy to understand why this strategy is such a failure; in fact, random vaccination gives the same weight to highly connected nodes as to nodes with low connectivity, and thus does not take in account the inhomogeneous connectivity property of the network. The result is that only complete immunization would fully stop the epidemic spreading and to achieve this it is necessary to ensure a high rate of immunization, such as the one shown in figure 6-8. In section 4.2 (page 117) it was already noted that the distribution of connectivities in this model follows a power law, as it happens in scale free networks, and also from the conceptual model, it is already implicit in the heterogeneous spatial distribution of population that there are different connectivities in the network (section 2.2.1 page 51) and therefore the valid explanations for scale free networks can be applied here.

Another property of scale free networks is the fact that their connectivity has a strong

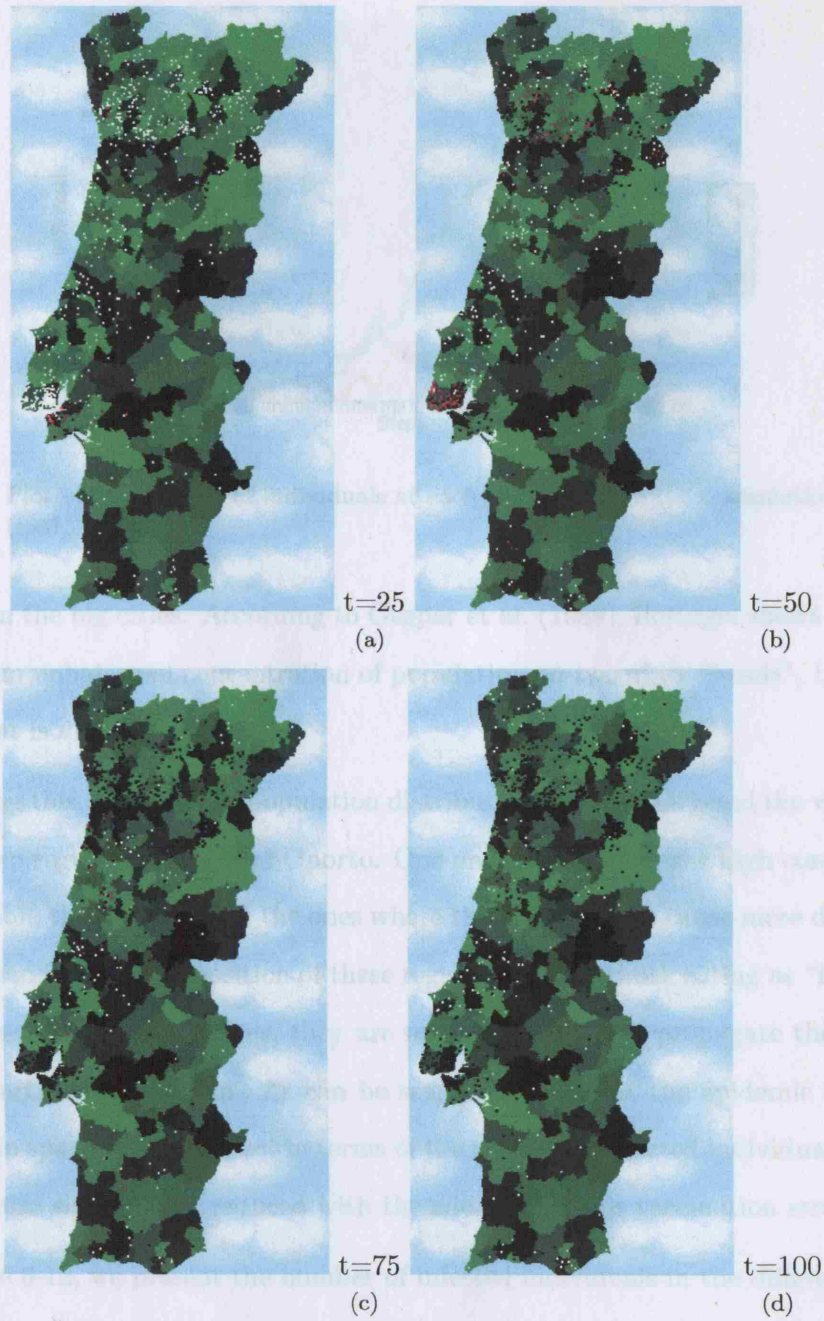


Figure 6-5: Screenshots of a simulation with ring vaccination. The vaccinated individuals are represented in blue.

resistance to random detachment of nodes, but is very sensitive to selective detachment of nodes. In other words, successful immunization strategies can be developed, taking advantage of the inhomogeneous connectivity property of these networks. In this context of target vaccination, a strategy can be focused in the highly connected areas, in

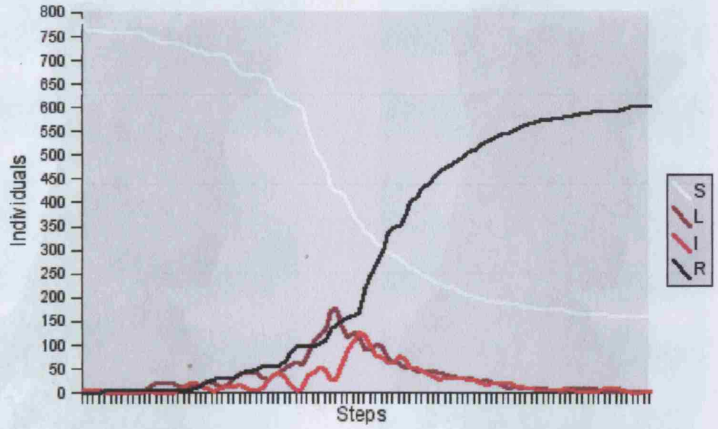


Figure 6-6: Plot of the number of individuals at each state over time in a simulation with ring vaccination (t=0,...,100).

this case, in the big cities. According to Gaspar et al. (1989), Portugal shows a "macrocephaly", an unbalanced concentration of population on two of its "heads", Lisbon and Oporto that is visible in figure 6-9.

Following this profile of the population distribution, we have targeted the vaccination on these two regions, Lisbon and Oporto. One one hand, due to the high concentration of population, these regions are the ones where the epidemic can cause more damage; on the other hand due to the position of these regions in the network acting as "hubs" with a greater number of connections, they are more susceptible to propagate the epidemic to other parts of the domain. As can be seen in figure 6-10, the epidemic is severely restricted in space and its impact in terms of the number of affected individuals in figure 6-11, was also significantly reduced with the adoption of this vaccination strategy.

In figure 6-12, we present the number of infected individuals in the different simulations which can provide a measure of the impact of the different strategies. The more the vaccination strategy manages to reduce the number of affected individuals when compared to the simulation with no vaccination, the greater its impact.

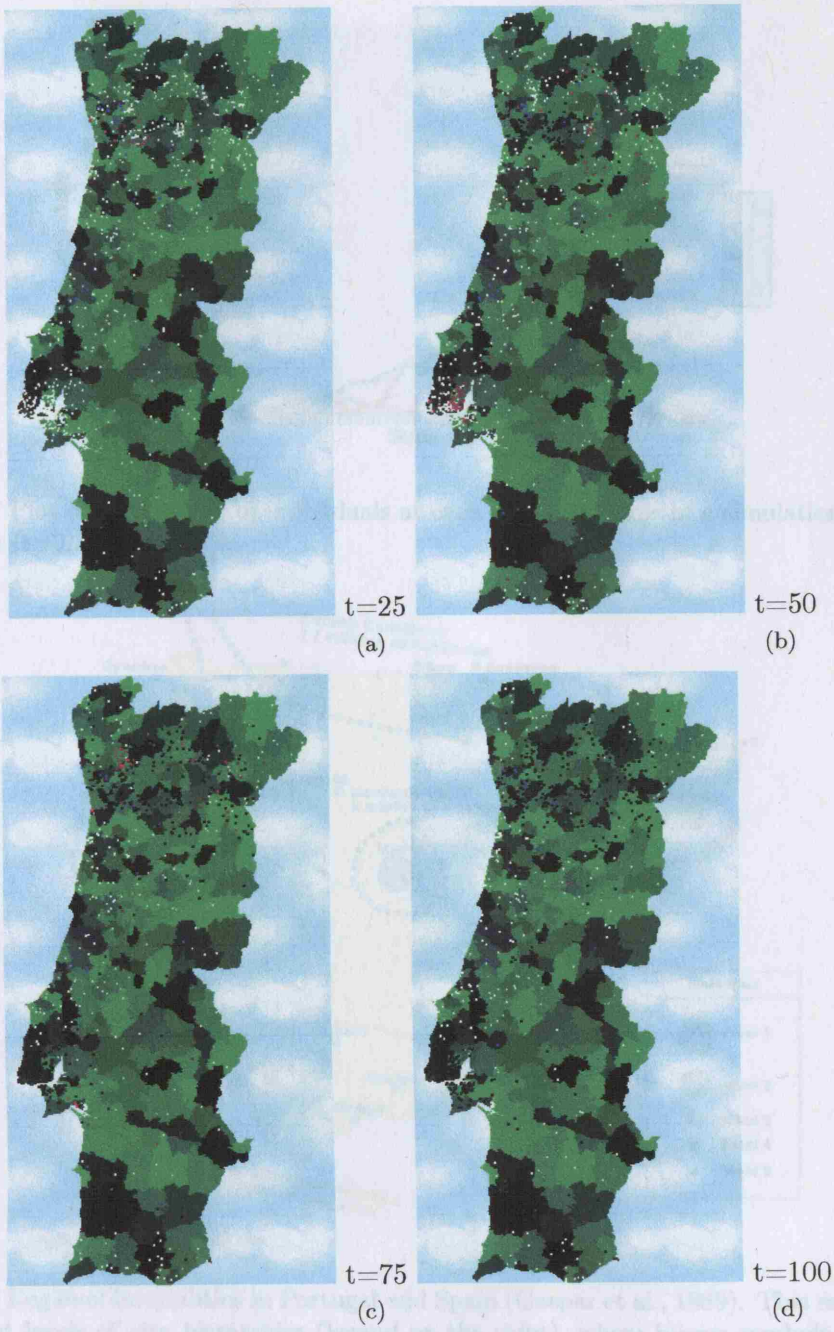


Figure 6-7: Screenshots of a simulation with mass vaccination.

6.3 Analysis of the Simulations

From this set of simulations, target vaccination seems to be the best strategy; however, too many uncontrolled conditions (or random forces) play a part in these simulations. Namely, the number of regions/individuals vaccinated differs in the three strategies and

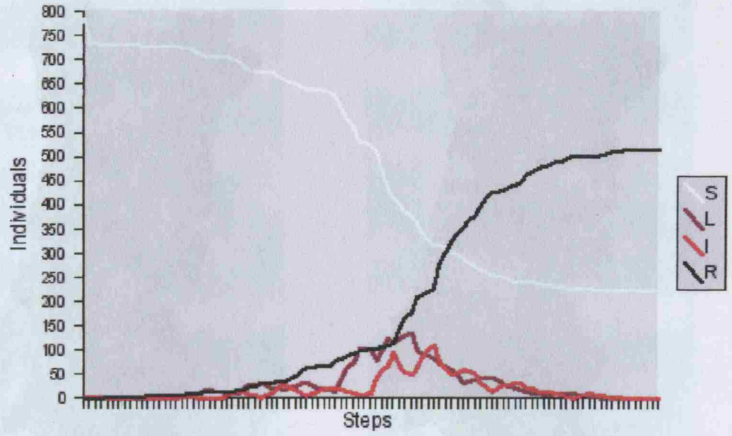


Figure 6-8: Plot of the number of individuals at each state over time in a simulation with mass vaccination (t=0,...,100).

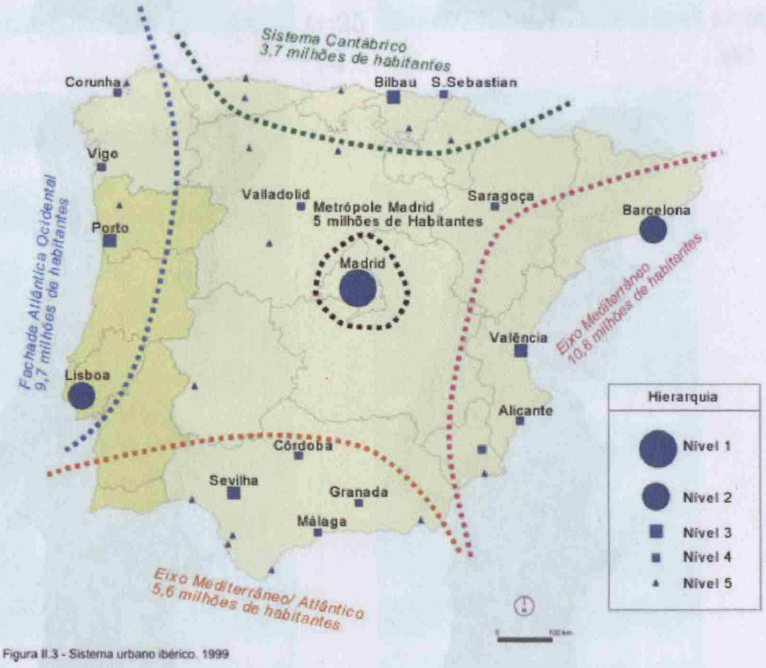


Figure 6-9: Regional inequalities in Portugal and Spain (Gaspar et al., 1989). This map presents the different levels of city hierarchies (legend on the right), where bigger symbols correspond to a higher level on the hierarchy scale. As we can see the hierarchic distribution in Portugal and Spain is very unbalanced with a small number of cities with a high hierarchic level, a high number of cities with a low hierarchic level and few cities in between these levels.

there are random features in the model (related to the probabilistic emergence of the movement network) that may create conditions for the development of clusters of infected individuals in some of the simulations. To prevent the first condition, it is possible to establish simulations, using as initial conditions, a constant number of vaccinated re-

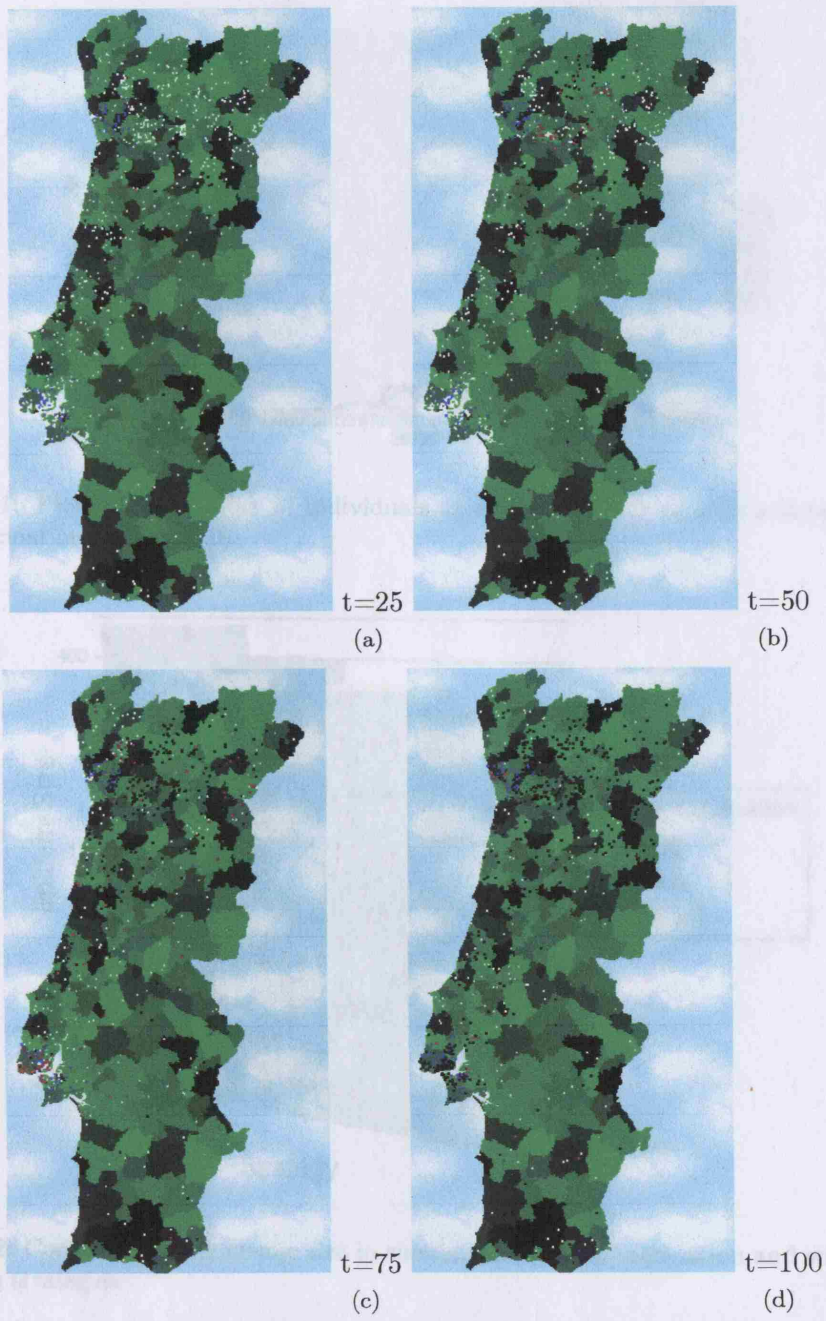


Figure 6-10: Screenshots of a simulation with target vaccination.

gions or a constant number of individuals; in this way, it is possible to isolate the spatial pattern of vaccination as the source of variability of the results. Concerning the second condition, nothing can be done to prevent the natural variability of the results, for this is not a deterministic model; however, the law of large numbers(Grimmett and Stirzaker, 1992) states that the average of a randomly selected large sample from a population is

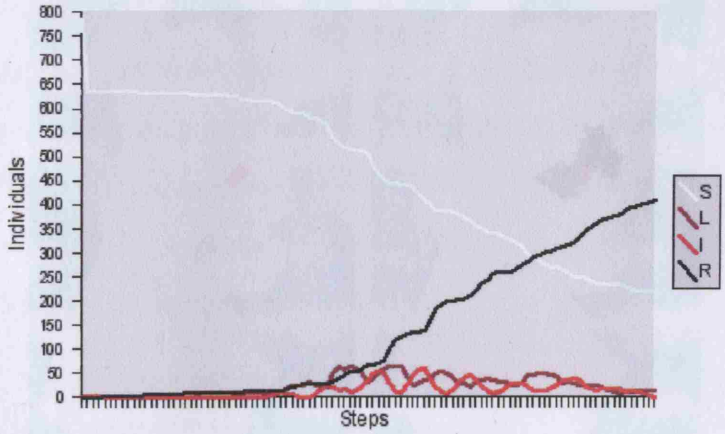


Figure 6-11: Plot of the number of individuals at each state over time in a simulation with target vaccination (t=0,...,100).

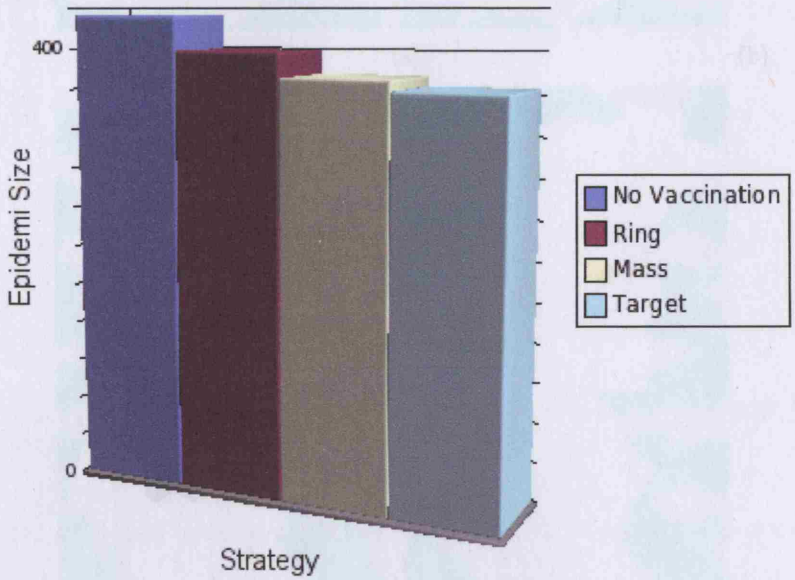


Figure 6-12: Comparison of epidemic size in simulations with no vaccination and with different vaccination strategies.

likely to be close to the average of the whole population, which encouraged us to run a larger number of simulations. Figure 6-13 shows the initial conditions for the simulations using a constant number of vaccinated regions. In this case, the optimization of vaccination involves areas and not the number of vaccines and this may be closer to reality if we think that in vaccination campaigns, displacement brings associated costs.

In this case, the number of vaccinated regions in each simulation is eight which corre-

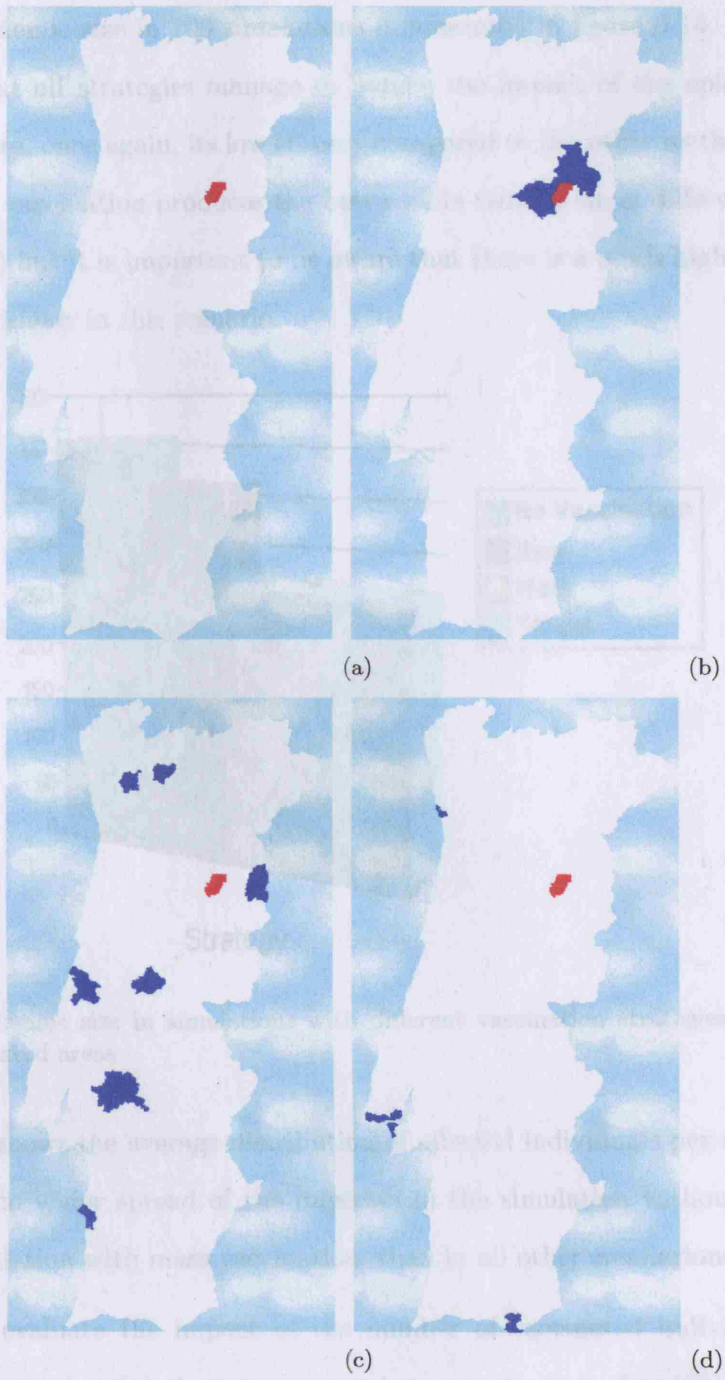


Figure 6-13: Screenshots of the initial conditions (t=0) for simulations with no vaccination and with different vaccinated strategies. In these simulations, although the strategies are different they use the same number of vaccinated areas.

sponds to different population densities according to the demographic distribution. As an indication, the number of vaccinated individuals is as follows: 17 in the ring strategy simulation, 33 in the mass strategy simulation and 100 in the target strategy simulation.

The average epidemic size in 100 simulations is presented in figure 6-14. One obvious conclusion is that all strategies manage to reduce the impact of the epidemic. Mass vaccination proves, once again, its low efficacy compared to the other methodologies. In this case, target vaccination produces the best results (with a small difference from the ring vaccination) but it is important to be aware that there is a much higher number of vaccinated individuals in this scenario.

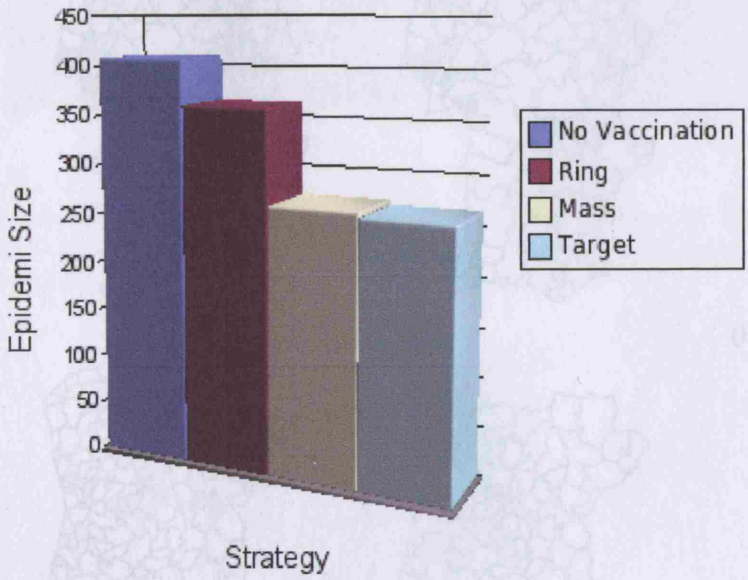


Figure 6-14: Epidemic size in simulations with different vaccination strategies but an equal number of vaccinated areas.

Figure 6-15 shows the average distribution of affected individuals per region. There is clearly a much wider spread of the infection in the simulation without vaccination and in the simulation with mass vaccination, than in all other simulations.

In order to evaluate the impact of the number of vaccinated individuals on the strategy results, some simulations were run using a constant number (28) of vaccinated individuals. In this way it is possible to isolate the spatial pattern of the strategy and understand better its effect on the results. The initial conditions for the ring, mass and target strategy simulations are shown in figures 6-16, 6-17 and 6-18.

In figure 6-19, we present a comparison of the average epidemic size in 100 simulations, under the previously referred initial conditions. In this case, the results are

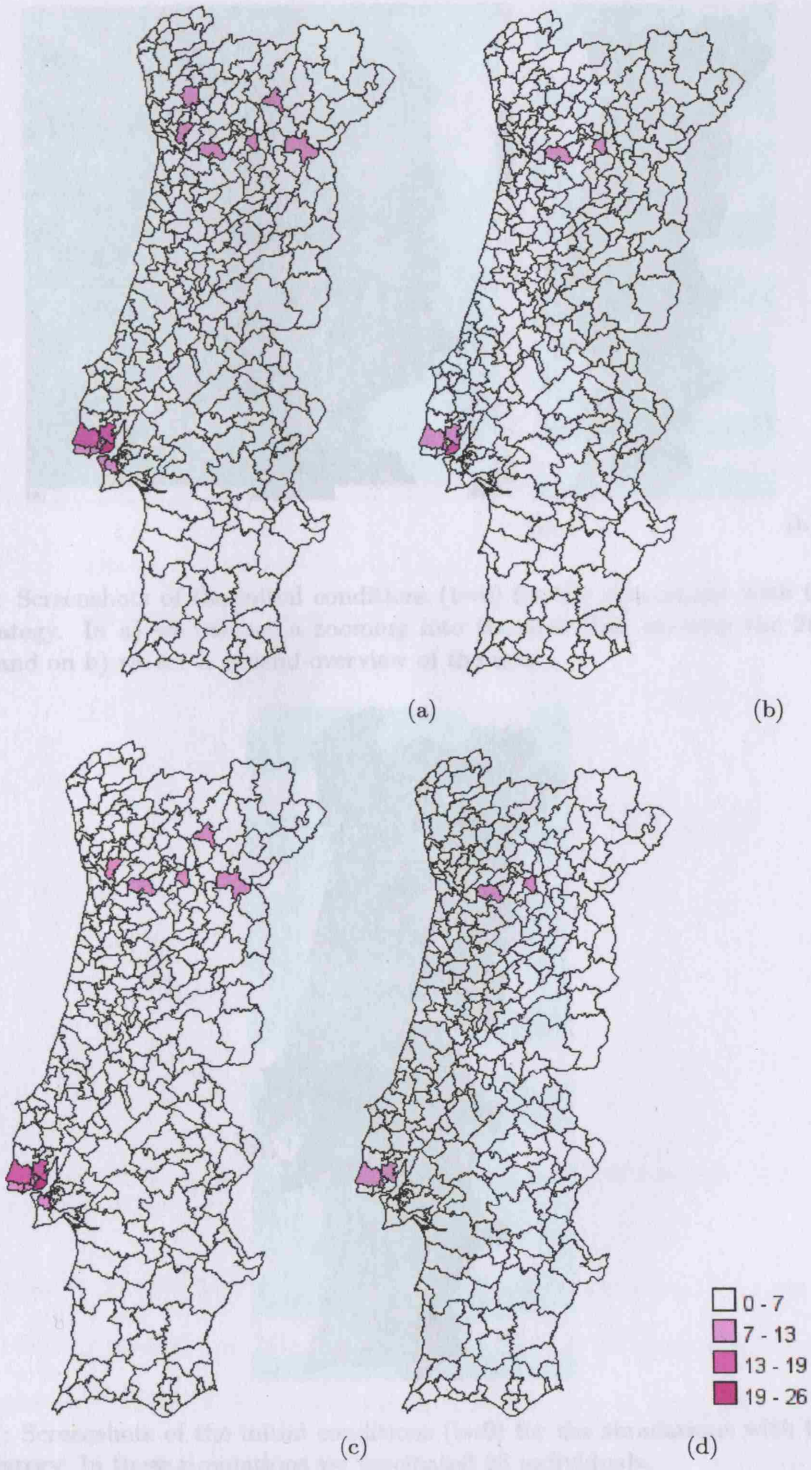
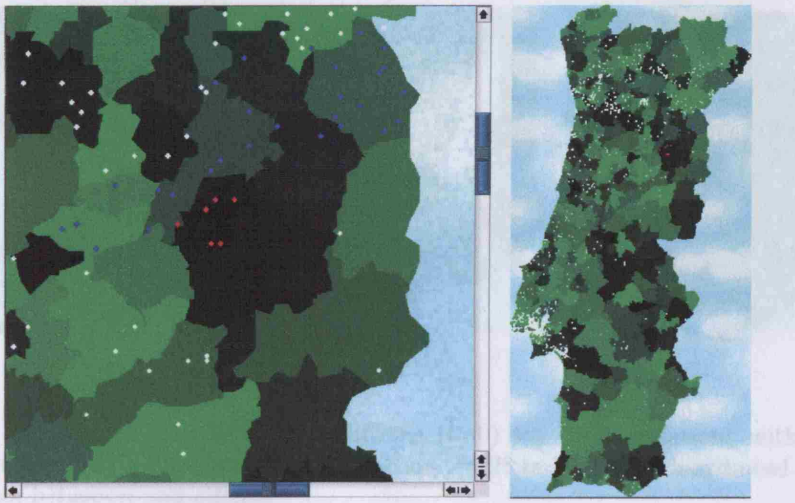


Figure 6-15: Average distribution of affected individuals in simulations with: a) no vaccination and (b) ring, c) mass and d) target strategies). Although the strategies are different, they use the same number of vaccinated areas.

significantly different from the ones with an equal number of vaccinated areas. Ring vaccination appears as the best strategy, followed by mass, and finally target vaccina-

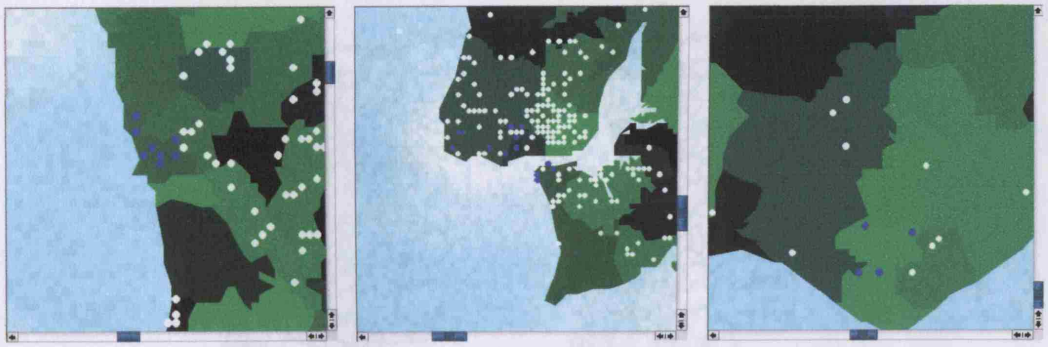


(a) (b)

Figure 6-16: Screenshots of the initial conditions (t=0) for the simulations with the ring vaccination strategy. In a) we can see a zooming into the area that encloses the 28 vaccinated individuals and on b) we see a general overview of the map.

Figure 6-17: Screenshots of the initial conditions (t=0) for the simulations with the mass vaccination strategy. In these simulations we vaccinated 28 individuals.

tion. One conclusion that may be drawn here is that in both cases, ring vaccination is quite effective in limiting the spread of the disease. This is because the infection is constrained straight away from its beginning before it reaches an outbreak. In the case



(a) (b) (c)

Figure 6-18: Screenshots of the initial conditions (t=0) for the simulations with the target vaccination strategy. In these simulations we vaccinated 28 individuals, distributed in the main cities a)Oporto, b)Lisbon and c)Faro.

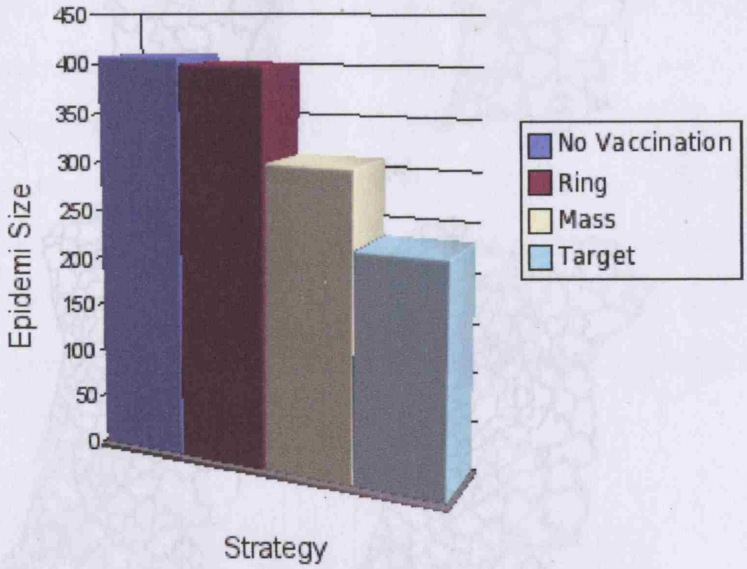


Figure 6-19: Epidemic size in simulations with different vaccination strategies but equal number of vaccinated individuals.

of target vaccination, the control of the epidemic only starts when it reaches the urban centres and that is already too late. In terms of networks, it is important to restrict the infection network while it is small, because a late action will imply the detachment of a higher number of nodes. The average spatial distribution of the epidemic is shown in figure 6-20 where is visible that the ring vaccination severely restricts the spread of the disease.

Based on the results of these simulations, it is possible to state that the infectious network can be restricted spatially through vaccination; if this restriction occurs earlier,

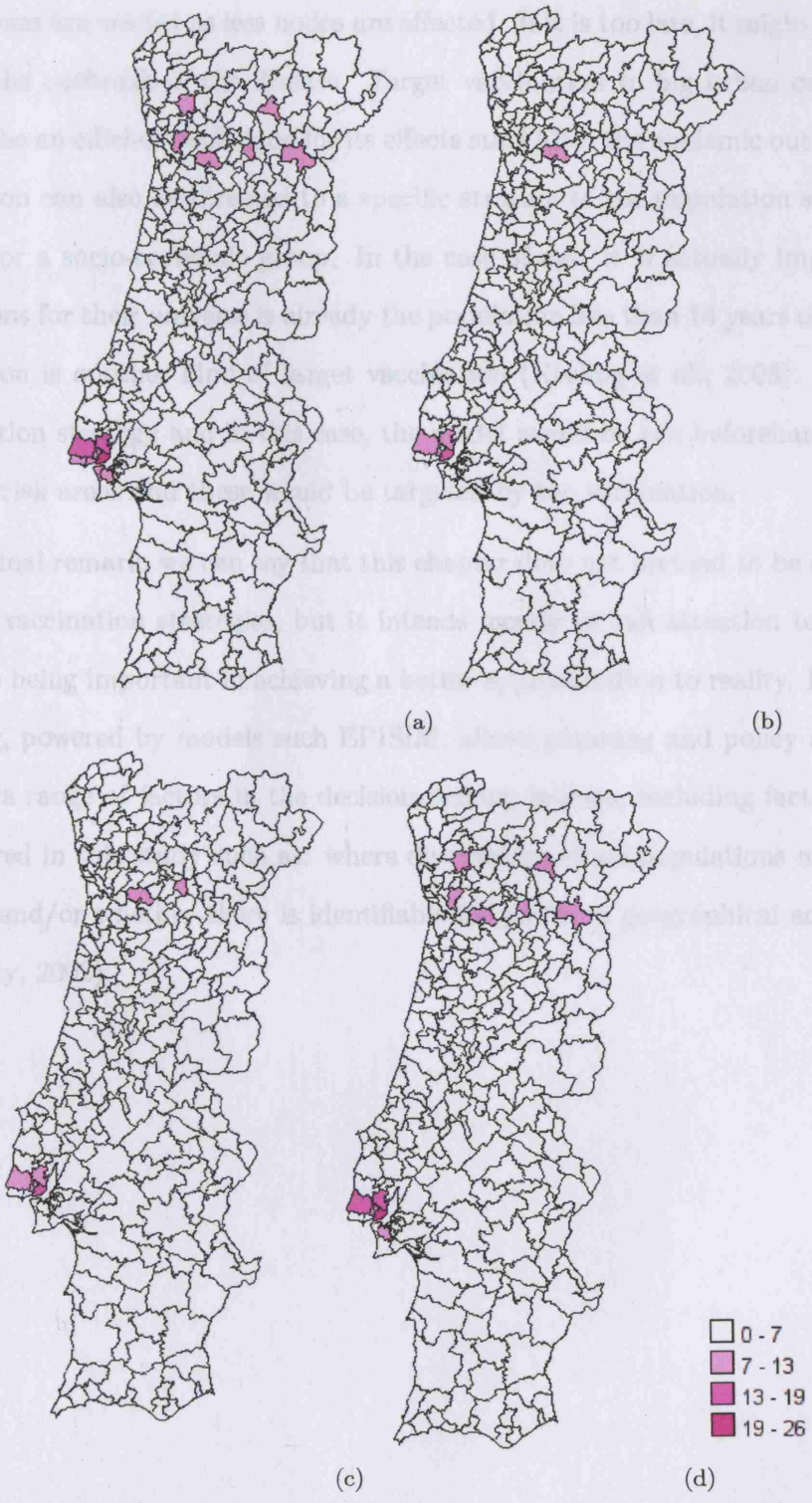


Figure 6-20: Average distribution of affected individuals in simulations with: a) no vaccination and (b) ring, c) mass and d) target strategies). Although the strategies are different they use the same number of vaccinated individuals.

less vaccines are needed as less nodes are affected. If it is too late, it might be impossible to stop the outbreak of the disease. Target vaccination to big urban centres did not prove to be an efficient technique for its effects start after the epidemic outbreak. Target vaccination can also be directed to a specific stratum of the population such as an age stratum or a socio-economic group. In the case of age, it is actually implicit in these simulations for their universe is already the population less than 14 years old. Predictive vaccination is another kind of target vaccination (Keeling et al., 2003). It is a smart optimization strategy and in this case, the model would be run beforehand to evaluate the high risk areas and these would be targeted by the vaccination.

As a final remark, we can say that this chapter does not pretend to be an exhaustive study of vaccination strategies, but it intends mostly to call attention to spatial variability as being important in achieving a better approximation to reality. In fact spatial modeling, powered by models such EPISIM, allows planning and policy authorities to consider a range of factors in the decision-making process, including factors that were not covered in this study such as: where disadvantaged subpopulations may be underserviced and/or whether there is identifiable inequities in geographical access (Brabyn and Skelly, 2002).

Chapter 7

Final Considerations

7.1 Conclusions

This work has proceeded through all steps of the modelling process. These are: to formulate a theoretical model, formulate a computational approach, implement it, analyze it and understand it, apply it to a real dataset and eventually to add other features. It is a long, interactive and iterative process with feedbacks from one stage to another. To the complexity of the phenomena in the study is added the complexity of programming. In fact, the programming itself was a major part of this project in terms of time spent and its importance to the simulation and user interaction. In ALife, the program is not only the translation of the experience but it is the experience itself, and so it makes sense to enhance this part of the study as more than a mere technical detail, one of the reasons why it deserved a whole chapter (chapter 3, page 75). In the end, and in the light of what has been done so far in literature, this work has brought attention to the spatial component of epidemics which is not yet a major focus in epidemiology. As a consequence, the incorporation of GIS concepts, principles and techniques in the core of the model should be emphasized (section 3.4, page 92). Considering the GIS data model, it is important to enhance the advantages of the vector approach in multi-scale models as it is the case of this one; as we show in figure 7-1, the vector representation of a complex geography (shown on the right) is a lot more realistic than in the initial CA-raster implementation (shown on the left).

The main strength of this model consists in the hybrid network/ABM approach which combines the functional perspective of networks with the spatial perspective of agent based models. This has proved to be a robust framework to approach the simulation of spatial epidemics, even in cases like this one where the social network is not so important for the spread of disease. The innovative contribution of the model is in the way it simulates movement and how it approaches the whole population-disease system, since the infection model itself, although relatively sophisticated, is not exactly new.

Once there is a working prototype, there is a temptation to continuously develop new functions, instead of focusing on the detailed analysis of existing ones. This was also the

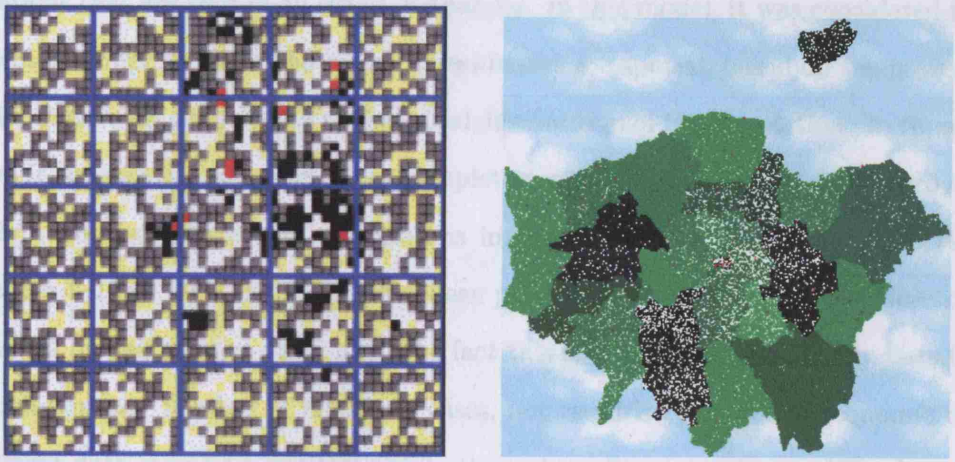


Figure 7-1: Examples of representations with raster and vector epidemic models.

case in this work and some components of the model, like the random mutation of the infection force and vaccination strategies, are not fully developed despite the fact they have been implemented. As these functionalities were being implemented we became aware that they could be further developed, and most of all considerable testing and analysis should be performed in order to make them useful for the model. In a thesis like this, there has to a compromise between the time dedicated to development and the time dedicated to analysis which is not always easy to establish. We can say there was an effort to *open new doors* and although exploring these new possibilities would be outside the scope of this study, some of these features would justify the entire thesis.

Similar to what happens in compartmental models, one temptation in this model would be to add more variables and components to the system. This could include: a characterization of the physical environment and infra-structures, like transport networks and urban centres, and adding more disaggregated details about the individuals such as socio-economic attributes. However, this would make the model more intensive and it would increase exponentially its complexity in terms of relations between variables; similar to many compartmental models, it would become slower and more difficult to analyze. The bottom-up approach to complexity relies on simple rules that allow the emergence of complex behaviors; the challenge is in capturing or synthesizing the essential aspects of the system that rule the mechanics of its behavior - that is, the

relations that are kept in all scales of analysis. In this model, it was considered that the essential components for the spread of epidemics are spatial, based on the movement of individuals with a sensitivity to the initial distribution of the population. In these terms, even if the movement model is not completely realistic, this approach has provided an insight into how movement mechanisms influence the spreading of the infection (see chapter 4, 116). This would not have been possible with a complicated socio-economic-physical model. On the other hand the fact this is a simple model, purely based on the population and number of infectious cases, not requiring any social-economic or environment data may be a positive point, as good quality datasets are difficult to obtain (see section 3.5, page 109). The more variables that are added, the more the model would be penalized by lack of quality data. Some strengths of the model have already been pointed out such as the bottom-up approach to epidemics, the representation of movement, the emphasis on the spatial component, the hybrid network-ABM approach and the GIS encapsulation. We are now going to focus on its weaknesses.

One important feature in this model is the view of space as a set of nested spatial scales (see section 2.2.1, on page 51). However, time is not scaled according to these levels which brings us to the question: **are these different movements happening on the same time scale? or could it be that neighborhood movement is happening on a shorter span than for instance, inter-region movement?** The answer to this question is probably "yes", and it would be helpful to find out how to incorporate this in the model which at first glance, is not an easy task. Also, we are aware that the whole concept lying underneath the movement model is based on a pure theoretical approach to the movement of people that would gain very much from a study with real data.

The question of the time scale does not become important until we simulate a case study which happens in chapter 5 (page 160). In this chapter, the latent and morbid periods are assigned for the particular disease using a correspondence between time steps and days (1 time step equals 1 day). However if we look at the span of the real outbreak (around 3 years), it is much longer than the outbreak we simulate (which is

around a month) which reveals an inconsistency suggesting that time is "contracted" in the model! This issue does not influence what we want to demonstrate in the sensitivity analysis in chapter 4 (page 116) but it is something that has to be considered if we want to work with real data.

Another sort of issue in this model is aggregation. Aggregation is something to avoid in ABMs or individual-based models. However due to technical limitations (which are explained in section 5.2, page 167), each individual was considered to actually represent a group of individuals which led to the neighborhood (and infection) radius representing the neighborhood of a group of individuals; the consequence of this is that the infection process is not peer-to-peer, but is actually a generalization of this. In pedestrian models, the relation between the agents and the streets is correct, and the number of agents reproduces the number of people; but to model larger areas with a greater number of agents, it is necessary either to increase massively the computation or to simplify. In this work, the second strategy has been followed. One consequence of this approach is that we cannot look at the results at an individual level. Although the dynamics of the model are (also) at a micro scale with the aggregation at the bottom, the results are only valid at a macro scale. This is an agent-based model with aggregated results which is not particularly unusual in the ALife approach where global patterns emerge from behavior at the bottom. It is also worth mentioning that there is a certain confusion between the map in the user interface and the geography in the model. This happens because all the programming of the model relies on geometrical objects in the map display. This approach, although providing an accurate representation of the model is not recommended since it limits the operations performed by the model in terms of what is possible to display. It might be the case that it is better to keep the detailed model and implement a map only with aggregated results like the ones we have illustrated in section 4.3 (page 137). These proved to give a clearer view of the data.

Finally, regarding the infection model, it is important to remark that, as it was mentioned earlier in section 4.3 (page 137), the coefficients which control respectively the transition to an infectious and a removed state (θ and α) were not assumed as

variables and so their influence in the infection model is unknown. In systems with self-emergent behavior, the only way to study the precise consequences of changes in the parameters is by simulation since there are no mathematical tools that can provide this information. On the other hand, it is impossible to simulate the entire combination of variables in the system so in the end we have to deal with some uncertainty about the model which does not occur in deterministic mathematical models.

7.2 Future Developments

Because a work like this is never completely finished, in the next paragraphs we will suggest some developments that could benefit the model, either overcoming some weaknesses we have already identified or extending its capabilities.

The problems with specifying initial conditions have already been discussed in section 5.2 (page 167). It is possible to reduce the bias introduced by the relation involving susceptible/infectives by introducing a larger number of individuals in the simulation which can be achieved using more powerful computer. As these simulations are so demanding in terms of computer power, one important improvement would be to implement a distributed version of the model, running in parallel on several processors; in this way it would be possible to simulate a large number of individuals and a large number of regions. Another critical issue in our assumption was ignoring the demographic turnover (section 2, page 49) which is partly justified by the short time span of the disease. The hypothesis of a closed population also led to not considering boundary conditions. This is equivalent to considering that the domain is an island with no contact with the exterior. Both these assumptions were retained for maintaining the simplicity of the model, but they are rather weak, especially in a world which is now global in terms of travel and trade. These assumptions should definitely be revised in extending the model to diseases with larger life spans, such as AIDS. As it is stated at the beginning of chapter 2 (page 49) this model was developed with a specific intention of simulating diseases that **spread only through physical proximity not**

requiring any specific ties between individuals. Thus we only considered a spatial mixing environment, **although other mixing environments may be implicit in the model**, as it does have activities which are associated with each range of movement. Nevertheless in developing this approach to other diseases with a stronger social component, it might be worth introducing other kinds of mixing environment such as households, working places, schools and so on. One very promising aspect of the model was the random mutation of the infection force (section 4.3, page 137). This aspect was incorporated so that attention could be brought to the fact that there may be evolution, or at least change, in the epidemic system, as this represents the interaction between living organisms: population and disease (virus, bacteria, etc). In fact, the results with mutation of the infection force encourage and support the study of spatial epidemics as complex adaptive systems rather than simply just complex systems. In this case, the interactions between individuals and the infectious disease would trigger a feedback reaction that is a self regulation which enables evolution (and in this case, learning). These adaptations include the immune response of individuals to the disease and the resistance of the disease to antibiotics. Genetic algorithms and genetic programming, may be ways of looking at this problem.

Finally, if there is work to do on the conceptual side, there is certainly a lot more on the programming side for software is never a finished product. There are bugs to remove and functions to optimize and the program should be tested with other *Shapefiles.* The software was built to be flexible concerning the input of data, but as it was mostly tested with the same dataset, we cannot guarantee it would work as well with other data. One other improvement would be involve the access to the database. In this study, there is no direct link between the program and the database which currently is based on a manual process of integrating the information through the attribute table of the *Shapefiles.* However, it is possible to implement real database connectivity through ODBC[41], a standard database access method. In this way, it would be possible to establish a query inside the program, relating the geographic and population information, and it would

[41] http://www.webopedia.com/TERM/O/ODBC.html

be easier and quicker to integrate information from other diseases, even from different DBMs (like for instance, MySQL).

Although the purpose of any model is to predict, this was more an exploratory work, rather than a data driven study interested in reproducing a certain event. There is the case study in chapter 5 (page 160) for we believe it is important to enforce the link with reality and application, but this is certainly not the main point of this study and its results should not be regarded has the main delivery of this thesis. Our focus is much more on demonstrating the importance of the spatial component and the use of GIS in epidemic models and health Geography. With this work we hope to have brought new ideas concerning how to understand the behavior of a phenomenon that has gained so much attention at the beginning of the new century, unfortunately through its damaging consequences. In figure 7-2, we can see an example of a disease that recently presented a worldwide spread and is regarded as a serious threaten to public health. We also hope to have extended methodological ideas for studying diffusion phenomena in general.

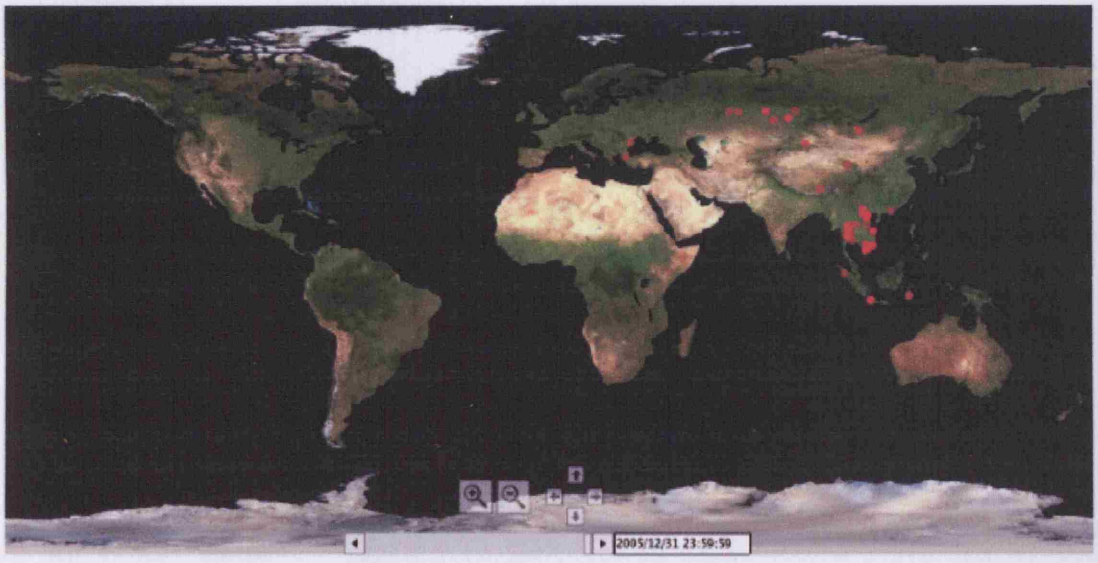


Figure 7-2: Avian Flu (H5N1) distribution on 2005/12/31 [42].

[42] http://brainoff.com/avianflu/

Finally, just some brief words to emphasize that this work provides insights into what is possible in research with free/open source tools which are at no cost to the developer and no cost to the user. As we see with a little bit more programming effort, it is possible

to develop a product that does not fall behind the standards of commercial solutions, that is in terms of user interface, GIS, data access, and so on. Through these chapters, there were already references to where to find some of the tools that can help in building this kind of software. As programs should be used, tested, and even transformed and this one will not finish its life in the context of this PhD thesis, it is already being implemented in a Windows version (adding to the original Linux one) thus allowing many more people to use it. A short term objective is to make this software available on *SourceForge*[43], where people will be able download it and have access to the code. In this way, we expect to extend the value of this contribution to future generations of researchers.

[43] http://sourceforge.net/

Bibliography

8th International Conference on GeoComputation (2005, August). *Can statistical methods on land-Use change patterns calibrate agent-based models?*, University of Michigan, Eastern Michigan University, USA. 8th International Conference on GeoComputation. `http://www.cscs.umich.edu/research/projects/sluce/publications/An_geocomp_%20abs.pdf`; accessed 23 August, 2006.

Alhir, S. S. (1998, September). *UML in a nutshell.* San Francisco, California, USA: O'Reilly.

Banks, R. B. (1994). *Growth and diffusion phenomena - mathematical frameworks and applications.* Berlin, Germany: Springer Verlag.

Batty, M. and B. Jiang (1999, April). Multi-agent simulation: New approaches to exploring space-time dynamics within gis. *CASA Working Paper Series* (10). `http://www.casa.ucl.ac.uk/publications/workingPaperDetail.asp?ID=10`; accessed 05 February, 2007.

Beynon-Davies, P. (1992). *Relational database design.* Oxford, UK: Blackwell Scientific Publications.

Bobrowski, S. (1997, September). *Oracle 8 architecture.* Osborne, USA: Oracle Press editions.

Booch, G. (1993, September). *Object-oriented analysis and design with applications.* Santa Clara, California: The Benjamin/Cummings Publishing Company, Inc.

Brabyn, L. and C. Skelly (2002, November). Modeling population access to New Zealand public hospitals. *International Journal of Health Geographics 1.* `http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=149398`; accessed 23 August, 2006.

Briss, P., L. Fehrs, R. Parker, and P. Wright (1994). Sustained transmission of mumps in a highly vaccinated population: assessment of primary vaccine failure and wanning vaccine induced immunity. *Journal of Infectious Diseases* (169), 77–82.

Brown, J. D. (1997). Skewness and kurtosis. *Shiken: JALT Testing & Evaluation SIG Newsletter 1*(1), 16–18. `http://www.jalt.org/test/bro_1.htm`; accessed 23 August, 2006.

Callaghan, A. (2005, january). Agent-based modelling applied to hiv/aids. *ERCIM NEWS (online edition)* (60), 50–51. `http://www.ercim.org/publication/Ercim_News/enw60/callaghan.html`; accessed 23 August, 2006.

Chalmers, D. J. (1996). *The Conscious Mind: In Search of a Fundamental Theory.* North Carolina, U.S.A.: Oxford Univ Pr.

Chen, S. (2001). Foot-and-mouth disease spread in this small world. Master's thesis, School of Cognitive and Computing Sciences, University of Sussex, Brighton, UK. `http://citeseer.ist.psu.edu/chen01footmouth.html`; accessed 23 August, 2006.

Chopard, B. and M. Droz (1998, April). *Cellular automata modelling of physical systems.* Collection Alea-Saclay: Monographs and Texts in Statistical Physics. Cambridge, UK: Cambridge University Press.

Cilliers, P. (1998). *Complexity and postModernism.* Connecticut, USA: London Routledge.

Cliff, A., P. Gould, A. Hoare, and N. Thrift (1995). *Diffusing geography: essays for Peter Haggett.* Oxford, UK; Cambridge, USA: Blackwell Publishers Inc.

Cliff, A., P. Hagget, J. Ord, and G. Versey (1981, April). *Spatial diffusion - An historical geography of epidemics in an island community.* Cambridge, UK: Cambridge University Press.

Cliff, A., P. Hagget, and M. Smallman-Raynor (1998, August). *Deciphering global epidemics - analytical approaches to the disease records of world cities, 1888-1912.* Cambridge Studies in Historical Geography. Cambridge, UK: Cambridge University Press.

Conference of the First Australian Conference on Artificial Life 2003 (2003, December). *Epidemic modelling using cellular automata*, Canberra, Australia. Conference of the First Australian Conference on Artificial Life 2003. `http://www.csse.uwa.edu.au/~scfu/research.html`; accessed 23 August, 2006.

Conolly, J. and M. Lake (2006, May). *Geographical information systems in archaeology.* Cambridge, UK: Cambridge University Press.

Crooks, A. (2006, September). Exploring cities using agent-based models and gis. *CASA Working Paper Series* (109). `http://www.casa.ucl.ac.uk/publications/workingPaperDetail.asp?ID=109`; accessed 05 February, 2007.

CUPUM (2005, June/July). *An integrated ABM and GIS model of infectious disease transmission*, London, UK. CUPUM.

Dauphiné, A. (1995). *Chaos, fractales et dynamiques en gographie.* Montepellier, France: Glip RECLUS.

Deijfen, M. (2000). Epidemics on social network graphs. Master's thesis, Department of Mathematics, Stockholm University, Stockholm University, SE 10691, Sweden. `http://www.math.su.se/~mia/uppsats.pdf#search=%22Epidemics%20on%20Social%20Network%20Graphs%22`; accessed 23 August, 2006.

DeMers, M. N. (2000). *Fundamentals of geographical information systems* (2 ed.). New York, USA: John Wiley & Sons, Inc.

Dias, J., M. Cordeiro, M. Afzal, and M. Freitas (1996). Mumps epidemic in portugal despite high vaccine coverage - preliminary report. Technical report, OPorto, Portugal.

Diekmann, O. and J. Heesterbeek (2000, February). *Mathematical epidemiology of infectious diseases.* New York, USA: John Wiley & Sons.

Epstein, J. M. (1997). *Nonlinear dynamics, mathematical biology and social science.* Santa Fe Institute Studies in the Sciences of Complexity Lecture Notes. Reading, MA, USA: Addison-Wesley.

Fuentes, M. and M. Kuperman (1999, May). Cellular automata and epidemiological models with spatial dependence. *Physica A 267*(3), 471–486.

Fuks, H. and A. T. Lawniczak (2001). Individual-based lattice model for spatial spread of epidemics. *Discrete Dynamics in Nature and Society 6*, 191. `http://http://www.citebase.org/cgi-bin/citations?id=oai:arXiv.org:nlin/0207048`; accessed 23 August, 2006.

Gaspar, J., D. Abreu, J. Ferro, and C. J. Butler (1989). *Portugal os proximos 20 Anos: ocupacao e a organizacao do espaco*, Volume 2. Lisboa, Portugal: Fundacao Calouste Gulbenkian.

Gonçalves, G., A. Araujo, and M. M. Cardoso (1998). Outbreak of mumps associated with poor vaccine efficacy. Technical Report 12, OPorto, Portugal.

Grimmett, G. R. and D. R. Stirzaker (1992). *Probability and Random Processes* (2 ed.). Oxford: Clarendon Press.

Hornsby, K. (2006). Spatial diffusion: conceptualizations and formalizations. `http://citeseer.ist.psu.edu/328405.html`; accessed 23 August, 2006.

Jordan, T. G., M. Domosh, and L. Rowntree (1994, January). *The Human Mosaic: A Thematic Introduction to Cultural Geography* (6 ed.). US: Harpercollins College Div.

J.P.Lewis and U. Neumann (2003, January). Performance of java versus c++. http://citeseer.ist.psu.edu/328405.html; accessed 23 August, 2006.

Kasturirangan, R. (1990, April). Multiple scales in small-world graphs. Technical report. http://arxiv.org/abs/cond-mat/9904055; accessed 23 August, 2006.

Keeling, W., May, Davies, and Grenfell (2003, January). Modelling vaccination strategies against foot-and-mouth disease. *Nature* (421), 136–142.

Langton, C. (Ed.) (1989). *Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems (ALIFE '87)*, Volume 6 of *Santa Fe Institute Studies in the Sciences of Complexity*, Reading, MA, USA. Addison-Wesley.

Lawson, A. B. (2001, July). *Statistical methods in spatial epidemiology.* Chichester, UK: John Wiley & Sons.

Light, S. O. (2000). *Better understanding.* eBookMall, Inc. http://www.ebookmall.com/authors/Stone_O_Light/; accessed 23 August, 2006.

Longley, P. A., M. F. Goodchild, D. J. Maguire, and D. W. Rhind (1999, February). *Geographical information systems - volume 1: principles and technical issues.* New York, USA: John Wiley & Sons, Inc.

Mansilla, R. and J. L. Gutierrez (2000). Deterministic site exchange cellular automata model for the spread of diseases in human settlements. *Bulletin of Mathematical Biology*, 1–12. http://www.citebase.org/abstract?id=oai:arXiv.org:nlin/0004012; accessed 23 August, 2006.

Margolus, T. T. N. (1987, April). *Cellular automata Machines: a new environment for modeling.* Cambridge, MA, USA: MIT Press.

Montgomery, S. L. (1994, March). *Object oriented information engineering.* New York, USA: Academic Press Inc.

Moore, C. and M. E. J. Newman (2000). Epidemics and percolation in small-world networks. Technical Report 00-01-002, Santa Fe Institute, Santa Fe Institute. 1399

Hyde Park Road, Santa Fe, New Mexico 87501. http://www.santafe.edu/sfi/publications/Working-Papers/00-01-002.pdf; accessed 23 August, 2006.

Neumann, J. V. (1996). *Theory of self-reproducing automata.* Urbana, Illinois, USA: University of Illinois.

Newman, M. (1999). Small worlds: the structure of social networks. Technical Report 99-12-080, Santa Fe Institute, Santa Fe Institute. 1399 Hyde Park Road, Santa Fe, New Mexico 87501. http://ideas.repec.org/p/wop/safiwp/99-12-080.html; accessed 23 August, 2006.

Nowak, A. and M. Lewenstein (1996, January). *Computer simulations from the philosophy of science point of view*, pp. 249–285. Dodrecht, Netherlands: Kluwer Academic Publishers. http://cognitrn.psych.indiana.edu/rgoldsto/complex/nowak.pdf; accessed 23 August, 2006.

Parallel Problem Solving from Nature - PPSN VIII: 8th International Conference (2004). *Spread of vector borne diseases in a population with spatial structure*, Birmingham, UK. Parallel Problem Solving from Nature - PPSN VIII: 8th International Conference.

Pastor-Satorras, R. and A. Vespignani (2002). Epidemics and immunization in scale-free networks. *Contribution to 'Handbook of Graphs and Networks:From the Genome to the Internet'.* http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0205260; accessed 23 August, 2006.

Quillin, M. J. (2001, November). Object oriented analysis and design: What is it? how does it work? why is it used? http://www.umsl.edu/~sauter/analysis/488_f01_papers/quillin.htm; accessed 05 February, 2007.

Reynolds, C. (1997, September). Individual-based models- an annotated list of links. http://www.red3d.com/cwr/ibm.html; accessed 23 August, 2006.

Reynolds, R., R. Whallon, and S. Goodhall (2001). Transmission of cultural traits by emulation: An agent-based model of group foraging behavior. *Journal of Memetics*

- *Evolutionary Models of Information Transmission* (4). http://jom-emit.cfpm.org/2001/vol4/reynolds_r&al.html; accessed 05 February, 2007.

Savitch, W. (1999). *Problem solving with C++: the object of programming.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Skansholm, J. (1997, June). *C++: from the beginning.* International Computer Science Series. Boston, MA, USA: Addison-Wesley Publishing Company.

Watt, A. and F. Policarpo (2000, December). *3D games - real time rendering and software technology.* New York, USA: Addison Wesley.

Watts, D. and S. Strogatz (1998, June). Collective dynamics of 'small world' networks. *Nature* (393), 440–442.

Weisstein, E. W. (2007, January). Graph. http://mathworld.wolfram.com/Graph.html; accessed 05 February, 2007.

Wolfram, S. (2002, May). *A new kind of science.* Illinois, USA: Wolfram Media INC.

Zheng, D., P. Hui, S. Trimper, and B. Zheng (2003, August). Epidemics and dimensionality in hierarchical networks. *Quantitative Biology cond-mat/0308502.* http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0308502; accessed 23 August, 2006.

CD-ROM about EpiSIM

The CD-ROM attached in end of this thesis contains additional material about EpiSIM, the software developed in the context of this PhD. Since the software is not platform independent we do not present it in the CD-ROM but we present a website instead, with an overview of its functionalities illustrated by screenshots and animations of the user interface. In this way, we hope to add more detail to the description of the program presented in chapter 3 (page 75).

The software was implemented in Linux using the GNU C++ compiler.